

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2003年 3月25日

出 願 番 号

Application Number:

特願2003-083243

[ST.10/C]:

[JP 2003-083243]

出 願 人

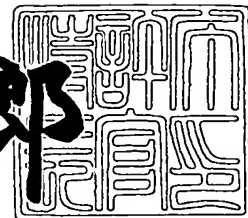
Applicant(s):

インターナショナル・ビジネス・マシーンズ・コーポレーション

2003年 5月20日

特 許 庁 長 官
Commissioner,
Japan Patent Office

太田 信一郎



出証番号 出証特2003-3037043

【書類名】 特許願
【整理番号】 JP9020233
【提出日】 平成15年 3月25日
【あて先】 特許庁長官 殿
【国際特許分類】 G06F 12/00
G06F 17/30

【発明者】

【住所又は居所】 神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ピー・エム株式会社 東京基礎研究所内

【氏名】 工藤 道治

【発明者】

【住所又は居所】 神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ピー・エム株式会社 東京基礎研究所内

【氏名】 村田 真

【発明者】

【住所又は居所】 神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ピー・エム株式会社 東京基礎研究所内

【氏名】 戸澤 晶彦

【発明者】

【住所又は居所】 神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ピー・エム株式会社 東京基礎研究所内

【氏名】 羽田 知史

【特許出願人】

【識別番号】 390009531

【氏名又は名称】 インターナショナル・ビジネス・マシーンズ・コーポレーション

【代理人】

【識別番号】 100086243

【弁理士】

【氏名又は名称】 坂口 博

【代理人】

【識別番号】 100091568

【弁理士】

【氏名又は名称】 市位 嘉宏

【代理人】

【識別番号】 100108501

【弁理士】

【氏名又は名称】 上野 剛史

【復代理人】

【識別番号】 100104880

【弁理士】

【氏名又は名称】 古部 次郎

【選任した復代理人】

【識別番号】 100118201

【弁理士】

【氏名又は名称】 千田 武

【手数料の表示】

【予納台帳番号】 081504

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9706050

【包括委任状番号】 9704733

【包括委任状番号】 0207860

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 情報処理装置、データベース検索システム及びそのアクセス権解析方法

【特許請求の範囲】

【請求項 1】 構造化文書で記述されたデータファイルを扱うデータベースへのアクセス権を解析する情報処理装置において、

前記データベースに対する検索条件を記述したパス式から検索オートマトンを生成する検索オートマトン生成部と、

アクセス制御規則を記述したアクセス制御ポリシーからアクセス制御オートマトンを生成するアクセス制御オートマトン生成部と、

前記検索オートマトン生成部にて生成された前記検索オートマトンおよび前記アクセス制御オートマトン生成部にて生成された前記アクセス制御オートマトンに関する論理演算を行って、前記パス式によるデータベース検索におけるアクセス権を判定する論理演算部と

を備えることを特徴とする情報処理装置。

【請求項 2】 前記データベースに格納されているデータファイルの構造を示すスキーマからスキーマオートマトンを生成するスキーマオートマトン生成部をさらに備え、

前記論理演算部は、前記スキーマオートマトン生成部にて生成された前記スキーマオートマトンを考慮して前記アクセス権の判定を行うことを特徴とする請求項 1 に記載の情報処理装置。

【請求項 3】 前記データベースに格納されているデータファイルのパスを記述したパステーブルを管理するパステーブル管理部をさらに備え、

前記スキーマオートマトン生成部は、前記パステーブル管理部に管理されている前記パステーブルから前記スキーマオートマトンを生成することを特徴とする請求項 2 に記載の情報処理装置。

【請求項 4】 前記データベースに対する検索方法を指定する問い合わせ式から前記パス式を抽出するパス式抽出部をさらに備えることを特徴とする請求項 1 に記載の情報処理装置。

【請求項 5】 前記問い合わせ式から抽出された個々の前記パス式に対する前記論理演算部によるアクセス権の判定結果に基づいて、前記問い合わせ式による前記データベース検索におけるアクセス権を判定する問い合わせ式アクセス権判定部をさらに備えることを特徴とする請求項 4 に記載の情報処理装置。

【請求項 6】 構造化文書で記述されたデータファイルを扱うデータベースへのアクセス権を解析する情報処理装置において、

前記データベースに格納されているデータファイルのパスを記述したパステーブルを管理するパステーブル管理部と、

前記データベースに対する検索条件を記述したパス式によって前記パステーブル管理部に管理されている前記パステーブル中の所定のパスを選択し、アクセス制御規則を記述したアクセス制御ポリシーを適用して、当該所定のパスに対する当該パス式によるデータベース検索におけるアクセス権を判定するアクセス権判定部と

を備えることを特徴とする情報処理装置。

【請求項 7】 前記データベースに対する検索条件を記述したパス式から検索オートマトンを生成する検索オートマトン生成部と、

前記アクセス制御規則を記述した前記アクセス制御ポリシーからアクセス制御オートマトンを生成するアクセス制御オートマトン生成部とをさらに備え、

前記アクセス権判定部は、前記検索オートマトン生成部にて生成された前記検索オートマトンを用いて前記所定のパスの選択を行い、前記アクセス制御オートマトン生成部にて生成された前記アクセス制御オートマトンを用いて前記所定のパスに対するアクセス権の判定を行うことを特徴とする請求項 6 に記載の情報処理装置。

【請求項 8】 前記データベースに対する検索方法を指定する問い合わせ式から前記パス式を抽出するパス式抽出部をさらに備えることを特徴とする請求項 6 に記載の情報処理装置。

【請求項 9】 前記問い合わせ式から抽出された個々の前記パス式に対する前記アクセス権判定部によるアクセス権の判定結果に基づいて、前記問い合わせ式による前記データベース検索におけるアクセス権を判定する問い合わせ式アク

セス権判定部をさらに備えることを特徴とする請求項 8 に記載の情報処理装置。

【請求項 1 0】 XML 文書を格納したデータベースと、

前記データベースに対する検索に用いられる検索条件を記述したパス式およびアクセス制御規則を記述したアクセス制御ポリシーに基づいて、当該パス式を用いた前記データベースの検索におけるアクセス権が、常に許可、常に拒否、不確定のいずれであるかを判定するアクセス権解析器と
を備えたことを特徴とするデータベース検索システム。

【請求項 1 1】 前記アクセス権解析器は、

前記データベースに対する検索条件を記述したパス式から検索オートマトンを生成する検索オートマトン生成部と、

前記アクセス制御規則を記述した前記アクセス制御ポリシーからアクセス制御オートマトンを生成するアクセス制御オートマトン生成部と、

前記検索オートマトン生成部にて生成された前記検索オートマトンおよび前記アクセス制御オートマトン生成部にて生成された前記アクセス制御オートマトンに関する論理演算を行って、前記パス式による前記データベース検索におけるアクセス権を判定する論理演算部と

を備えることを特徴とする請求項 1 0 に記載のデータベース検索システム。

【請求項 1 2】 前記データベースに対する検索方法を指定する問い合わせ式から前記パス式を抽出するパス式抽出部と、

前記問い合わせ式から抽出された個々の前記パス式に対する前記論理演算部によるアクセス権の判定結果に基づいて、前記問い合わせ式による前記データベース検索におけるアクセス権を判定する問い合わせ式アクセス権判定部と
をさらに備えることを特徴とする請求項 1 1 に記載のデータベース検索システム。

【請求項 1 3】 前記アクセス権解析器は、

前記データベースに格納されているデータファイルのパスを記述したパステーブルを管理するパステーブル管理部と、

前記データベースに対する検索条件を記述したパス式によって前記パステーブル管理部に管理されている前記パステーブル中の所定のパスを選択し、前記アク

セス制御規則を記述した前記アクセス制御ポリシーを適用して、当該所定のパスに対する当該パス式によるデータベース検索におけるアクセス権を判定するアクセス権判定部と

を備えることを特徴とする請求項10に記載のデータベース検索システム。

【請求項14】 前記データベースに対する検索方法を指定する問い合わせ式から前記パス式を抽出するパス式抽出部と、

前記問い合わせ式から抽出された個々の前記パス式に対する前記アクセス権判定部によるアクセス権の判定結果に基づいて、前記問い合わせ式による前記データベース検索におけるアクセス権を判定する問い合わせ式アクセス権判定部とをさらに備えることを特徴とする請求項13に記載のデータベース検索システム。

【請求項15】 コンピュータを用いてXML文書を格納したデータベースへのアクセス権を解析するアクセス権解析方法であって、

前記データベースに対する検索条件を記述したパス式から検索オートマトンを生成し、アクセス制御規則を記述したアクセス制御ポリシーからアクセス制御オートマトンを生成し、生成された当該検索オートマトンおよび当該アクセス制御オートマトンを所定の記憶手段に格納するステップと、

前記所定の記憶手段に格納された前記検索オートマトンおよび前記アクセス制御オートマトンに関する論理演算を行って、前記データベースに格納された前記XML文書を調べることなく、前記パス式によるデータベース検索におけるアクセス権を判定するステップと

を含むことを特徴とするアクセス権解析方法。

【請求項16】 コンピュータを用いてXML文書を格納したデータベースへのアクセス権を解析するアクセス権解析方法であって、

前記データベースに対する検索条件を記述したパス式によって、所定の記憶手段に格納され前記データベースに格納されているデータファイルのパスを記述したパステーブルから所定のパスを選択するステップと、

アクセス制御規則を記述したアクセス制御ポリシーを適用して、前記データベースに格納された前記データファイルを調べることなく、前記所定のパスに対す

る前記パス式によるデータベース検索におけるアクセス権を判定するステップとを含むことを特徴とするアクセス権解析方法。

【請求項 17】 コンピュータを制御して、構造化文書で記述されたデータファイルを扱うデータベースへのアクセス権を解析するプログラムであって、

前記データベースに対する検索条件を記述したパス式から検索オートマトンを生成する検索オートマトン生成手段と、

アクセス制御規則を記述したアクセス制御ポリシーからアクセス制御オートマトンを生成するアクセス制御オートマトン生成手段と、

生成された前記検索オートマトンおよび前記アクセス制御オートマトンに関する論理演算を行って、前記パス式によるデータベース検索におけるアクセス権を判定する論理演算手段として

前記コンピュータを機能させることを特徴とするプログラム。

【請求項 18】 前記データベースに対する検索方法を指定する問い合わせ式から前記パス式を抽出するパス式抽出手段と、

前記問い合わせ式から抽出された個々の前記パス式に対するアクセス権の判定結果に基づいて、前記問い合わせ式による前記データベース検索におけるアクセス権を判定する問い合わせ式アクセス権判定手段として

前記コンピュータをさらに機能させることを特徴とする請求項 17 に記載のプログラム。

【請求項 19】 コンピュータを制御して、構造化文書で記述されたデータファイルを扱うデータベースへのアクセス権を解析するプログラムであって、

前記データベースに格納されているデータファイルのパスを記述したパステーブルを管理するパステーブル管理手段と、

前記データベースに対する検索条件を記述したパス式によって前記パステーブル管理手段に管理されている前記パステーブル中の所定のパスを選択し、アクセス制御規則を記述したアクセス制御ポリシーを適用して、当該所定のパスに対する当該パス式によるデータベース検索におけるアクセス権の有無を判定するアクセス権判定手段として

前記コンピュータを機能させることを特徴とするプログラム。

【請求項 2 0】 前記データベースに対する検索方法を指定する問い合わせ式から前記パス式を抽出するパス式抽出手段と、

前記問い合わせ式から抽出された個々の前記パス式に対するアクセス権の判定結果に基づいて、前記問い合わせ式による前記データベース検索におけるアクセス権を判定する問い合わせ式アクセス権判定手段として
前記コンピュータをさらに機能させることを特徴とする請求項 1 9 に記載のプログラム。

【請求項 2 1】 請求項 1 7 乃至請求項 2 0 のいずれかに記載のプログラムを、コンピュータが読み取り可能に記録した記録媒体。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、アクセス制御に関し、特に XML (Extensible Markup Language) で記述された文書ファイルを扱うデータベースのためのアクセス制御に関する。

【0 0 0 2】

【従来の技術】

XML で記述された文書 (XML 文書) は、要素と属性からなる木構造にて表現することができる。この木構造についての条件をスキーマによって記述することができる。したがって、このスキーマを用いることにより、特定の条件を満たす木構造をもつ XML 文書のみを許容するようなシステムを作ることができる (スキーマについては、例えば、非特許文献 1 参照)。

XML 文書を扱うデータベース (XML データベース) では、文書を取り出すだけではなく、XML 文書中の要素や属性を限定して選択的に取り出すことができる。例えば、段落を表す要素 (< p > . . . < / p > で表現される要素) を全て取り出すといった処理が可能である。検索条件を記述する方法としては、X P a t h という W 3 C (World Wide Web Consortium) 勧告が広く用いられている。

【0 0 0 3】

また、XML データベースでは、単に要素や属性を検索するだけではなく、取

り出された要素や属性を用いて、新たなXML文書を作り出すこともできる。そのような機能を備えた検索言語として、XQueryという仕様がW3Cで設計されつつある。XQueryは、要素や属性を取り出すための機構としてXPathを用いる。

【0004】

さらにXMLデータベースにおいて、XML文書を構成する要素や属性に対してアクセスを行う必要がある場合について、具体例として、ある企業の従業員の一覧表を表現したXML文書を考える。各従業員については、その年収や社員番号が示されている。従業員、年収、社員番号は、それぞれemployee要素、salary要素、number要素によって表現されるものとする。このようなXML文書を格納したXMLデータベースでは、number要素についてのアクセスは特に制限する必要はないが、salary要素についてのアクセスは一部の人に限定しなければならないという場合がある。誰がどの要素・属性をアクセスできるかについての記述を、アクセス制御ポリシーと呼ぶ。ユーザは、アクセス制御ポリシーを記述して、XMLデータベースに与える。XMLデータベースは、XML文書に対するアクセス要求に対して、与えられたアクセス制御ポリシーを利用して、要素・属性へのアクセスを許可するのか拒否するのかを判定する。

【0005】

アクセス制御ポリシーを表現するための言語にXACL (XML Access Control Language) がある (XACLについては、例えば、非特許文献2参照)。XACLの言語仕様には、所定のアクセス要求に対してアクセス許可・拒否を決めるアルゴリズムが記述されている。このアルゴリズムは、所定の一つのノードに対する判定を行う。したがって、XPathもしくはXPathによる検索が複数のノードをアクセスする場合には、どのノードに対してもこのアルゴリズムを一度ずつ実行する。

【0006】

【非特許文献1】

村田真著、「XML [I] - XML SchemaとRelax -」、電子情報通信学会

誌、2001年、12月、Vol. 84、No. 12、p. 890-894

【非特許文献2】

工藤道治 (Michiharu Kudo)、羽田知史 (Satoshi Hada) 著、“XML Document Security based on Provisional Authorization”、Proceedings of the 7th ACM Conference on Computer and Communications Security、2000年、11月、p. 87-96

【非特許文献3】

J. ホップクロフト・J. ウルマン著 (野崎・高橋・町田・山崎訳)、「オートマトン 言語理論 計算論 I、II」、サイエンス社、1986年

【非特許文献4】

“XQuery 1.0: An XML Query Language”、[online]、2002年4月30日、W3C Working Draft、[平成15年1月29日検索]、インターネット
<URL: HYPERLINK "http://www.w3.org/TR/2002/WD-xquery-20020430/" http://www.w3.org/TR/xquery/>

【0007】

【発明が解決しようとする課題】

上述したように、XMLデータベースでは、アクセス要求に対して、与えられたアクセス制御ポリシーを利用して、要素・属性へのアクセスを許可するのか拒否するのかを判定する必要があるが、このXMLデータベースの性能を向上させるには、アクセス権限の判定を高速に行うことが必要となる。

しかし、アクセス制御ポリシーを表現する言語であるXACLの言語仕様に示されたアクセス権判定アルゴリズムは高速とは言えない。しかも上述のように、実際のアクセス権判定では、このアルゴリズムによる処理が多くのノードに対して繰り返されることとなるため、その実行性能は実用的とは言えない。

【0008】

なお、上記従来技術では、XMLデータベースの検索条件の記述方法としてXPath及びXQueryを挙げたが、このXPathに類似するもの（パス式と呼ぶ）やXQueryに類似するもの（問い合わせ式と呼ぶ）は他にも存在する。しかし、これらによるXMLデータベースは皆、

- ・パス式 (path expression) (又はパス式を含む問い合わせ式)
- ・XML文書 (XML document)

を与えられて

- ・XML文書中のノード (要素・属性・テキスト)

を検索する。そして、検索されたノードに対するアクセスが許されているかどうかを

- ・アクセス制御ポリシー (access control policy)

と照合して判定する。

といった手順で処理を実行する点は同様である。したがって、いずれの方法であっても、例えば検索された要素・属性が1000個あれば、この判定も1000回繰り返されるため、この処理に多大な時間を要することは、共通する問題点と言える。

【0009】

そこで本発明は、XML等の構造化文書で記述されたデータファイルを扱うデータベースにおいて、データファイル自体やそのノードを調べることなくアクセス権解析 (access rights analysis) を行うことを実現し、データベースの検索性能を向上させることを目的とする。

【0010】

【課題を解決するための手段】

上記の目的を達成する本発明は、XMLのような構造化文書で記述されたデータファイルを扱うデータベースへのアクセス権を解析する、次のように構成された情報処理装置として実現される。この情報処理装置は、データベースに対する検索条件を記述したパス式から検索オートマトンを生成する検索オートマトン生成部と、アクセス制御規則を記述したアクセス制御ポリシーからアクセス制御オートマトンを生成するアクセス制御オートマトン生成部と、データベースに格納されているデータファイルの構造を示すスキーマからスキーマオートマトンを生成するスキーマオートマトン生成部と、生成された検索オートマトン、アクセス制御オートマトン及びスキーマオートマトンに関する論理演算を行って、かかるパス式によるデータベース検索におけるアクセス権を判定する論理演算部とを備

えることを特徴とする。

【0011】

また、本発明の情報処理装置は、データベースに格納されているデータファイルのパスを記述したパステーブルを管理するパステーブル管理部をさらに備え、このパステーブルから直接、もしくはスキーマオートマトンを生成して、論理演算部がアクセス権を判定する構成とすることができる。

さらに、データベースに対する検索方法を指定する問い合わせ式からアクセス権解析の対象となるパス式を抽出することができる。この場合、問い合わせ式から抽出された個々のパス式に対する論理演算部によるアクセス権の判定結果に基づいて、問い合わせ式によるデータベース検索におけるアクセス権を判定する。

【0012】

さらにまた、上記の目的を達成する他の本発明は、コンピュータを用いてXML文書を格納したデータベースへのアクセス権を解析する、次のようなアクセス権解析方法としても実現される。このアクセス権制御方法は、データベースに対する検索条件を記述したパス式から検索オートマトンを生成し、アクセス制御規則を記述したアクセス制御ポリシーからアクセス制御オートマトンを生成し、生成された検索オートマトンおよびアクセス制御オートマトンを所定の記憶手段に格納するステップと、これら検索オートマトンおよびアクセス制御オートマトンに関する論理演算を行って、データベースに格納されたXML文書を調べることなく、パス式によるデータベース検索におけるアクセス権を判定するステップとを含むことを特徴とする。

【0013】

また、本発明の他のアクセス権解析方法は、データベースに対する検索条件を記述したパス式によって、所定の記憶手段に格納され、データベースに格納されているデータファイルのパスを記述したパステーブルから所定のパスを選択するステップと、アクセス制御規則を記述したアクセス制御ポリシーを適用して、データベースに格納されたデータファイルを調べることなく、所定のパスに対するパス式によるデータベース検索におけるアクセス権を判定するステップとを含むことを特徴とする。

【 0 0 1 4 】

さらに本発明は、コンピュータを制御して上述したアクセス権解析器あるいはデータベース検索システムの各機能を実現させるプログラム、またはコンピュータに上記アクセス権解析方法の各ステップに相当する処理を実行させるプログラムとしても実現される。このプログラムは、磁気ディスクや光ディスク、半導体メモリ、その他の記録媒体に格納して配布したり、ネットワークを介して配信したりすることにより、提供される。

【 0 0 1 5 】

【発明の実施の形態】

以下、添付図面に示す実施の形態に基づいて、この発明を詳細に説明する。

本発明は、XMLデータベースに対するアクセス権解析において、XML文書の構造を調べて行う実際の解析に先立って、スキーマもしくはパステブルを用いることにより予備的なアクセス権解析を実行する。この予備的な解析において、検索条件を記述する所定のパス式によるXMLデータベースへのアクセスが、常に許可される (always permitted) か、常に拒否される (always denied) か、またはアクセス可否が不確定 (indeterminate) であるかを、XML文書の検索前に判定する。そして、アクセス可否が不確定と判定した場合にのみ、XML文書の構造を調べて行う通常のアクセス権解析を行う。

【 0 0 1 6 】

検索条件を記述したパス式とアクセス条件を示すアクセス制御ポリシーとの組合せによっては、XML文書の検索前の段階で、アクセス権の解析が可能な場合がある。解析可能な例をいくつか挙げる。

例 1 : アクセス制御ポリシーは、ユーザ `guest` が要素 `p` にアクセスすることを許しているものとする。XPath 式 (パス式) は `//p` とする。これは、タグ名 `p` を持つ要素すべてを取り出す。取り出された要素は、ユーザ `guest` によるアクセスが可能ながあらかじめ保障されている。どんな文書であっても、アクセス権のない要素を `//p` が取り出すことはあり得ない。この場合、1 つ 1 つの要素 `p` に対してアクセス権判定をする必要はない。

例2：逆に、要素 p へのユーザ $guest$ によるアクセスが禁止されているとする。このとき、XPath式 $//p$ が探し出す要素へのアクセス権をユーザ $guest$ が持たないことは明らかである。この場合、 $//p$ で要素を探し出すことすら必要ではない。単に、アクセス権がないとみなして処理すれば良い。

例3：アクセス制御ポリシーが、 sec 要素の子孫である p 要素へのアクセスを許しているとする。XPath式は $//sec//p$ とする。これは、タグ名 p を持つ要素であって、タグ名 sec を持つ要素の子孫であるもの全てを取り出す。この場合、取り出された要素に、ユーザ $guest$ がアクセス可能なことは明らかである。

例4：アクセス制御ポリシーは、例3と同様とする。そして、スキーマが存在し、このスキーマは、タグ名 p を持つ要素を、タグ名 sec を持つ要素の子要素としてしか許さないものとする。この場合、XPath式 $//p$ が取り出す要素に、ユーザ $guest$ によるアクセスが可能なことは明らかである。

例5：アクセス制御ポリシーが、 sec 要素およびその子孫へのアクセスを禁止しているものとする。XPath式は $//sec//p$ とする。このXPath式が探し出す要素へのアクセス権がないことは明らかである。

例6：アクセス制御ポリシーは、例5と同様とする。そして、スキーマが存在し、このスキーマは、タグ名 p を持つ要素を、タグ名 sec を持つ要素の子要素としてしか許さないものとする。この場合も、XPath式 $//p$ が探し出す要素へのアクセス権がないことは明らかである。

【0017】

これらは、XML文書やそのノードを調べることなくアクセス権解析が可能な場合の一例に過ぎない。このような予備的なアクセス権解析によりアクセス判定

が可能な場合は、XML文書の検索前にこの判定を一度だけ行えば良く、検索されたXML文書のノードごとにアクセス判定を繰り返す必要がないため、大幅な性能向上を望むことができる。

以下では、パス式 (X P a t h) を用いたアクセス権解析 (実施の形態 1、2)、パス式を内包する問い合わせ式 (X Q u e r y) を用いたアクセス権解析 (実施の形態 3)、これらの実装例 (実施の形態 4、5) に分けて本発明を説明する。

【 0 0 1 8 】

〔実施の形態 1〕

本実施の形態によるアクセス権解析では、アクセス制御ポリシー (access control policy) からアクセス制御オートマトン (access control automaton) Γ を生成し、スキーマ (schema) からスキーマオートマトン (schema automaton) S を生成する。そして、アクセス制御オートマトン Γ とスキーマオートマトン S とを用いて、所定のパス式 q に対しアクセス可否を判定する。

オートマトン (automaton) は、いくつかのシンボルから構成された列を調べて、受理されるかどうかを判定する機構として広く知られている。オートマトンの定義やオートマトンに対する論理演算については、例えば非特許文献 3 に、詳細に記述されている。また、シンボル列の集合を記述する方法として、正規表現 (regular expression) がある。正規表現は、オートマトンに変換することができ、その方法も、非特許文献 3 に記述されている。

【 0 0 1 9 】

ここで、パスとは、

- ・ XML 文書のルート要素からある要素 (木構造の葉でなくてもよい) に至るまでのタグ名を並べたもの

例：／ a ／ b ／ c (ここで、a、b、c、はタグ名)

- ・ XML 文書のルート要素からある要素 (木構造の葉でなくてもよい) に至るまでのタグ名を並べ、この要素のある属性の属性名を追加したもの

例：／ a ／ b ／ c ／ @ d (ここで、d は属性名)

- ・ XML 文書のルート要素からある要素 (木構造の葉でなくてもよい) に至るま

でのタグ名を並べ、この要素のテキストを参照するという情報を追加したもの

例：／a／b／c／text（）

のいずれかとする。

パス式 (path expression) とは、パスについての条件を表す式である。例えば、「ルート要素fooから出発して要素pに至る」は、パス式である。パス式の1つであるXPathでは、この条件式を／foo／／pと表現する。本実施の形態では、このXPathをパス式として用いる。

【0020】

アクセス制御ポリシーは、いくつかのアクセス制御規則 (access control rule) からなる。各規則は、パス式を用いてアクセスの可否を記述する。あるノードのパスが、このパス式に合致するとき、アクセスは許可または拒否される。上述のように、本実施の形態では、XML文書中のノードに対するアクセス権を判定するために、アクセス制御オートマトンを導入する。アクセス制御オートマトンとは、XML文書のルート要素からあるノードまでのパスが受理されるなら、またそのときに限ってアクセス権があるようなオートマトンである。

さらに本実施の形態では、XML文書中のノードをXPathによって取り出すために、検索オートマトン (query automaton) を導入する。検索オートマトンとは、XML文書のルート要素からあるノードまでのパスが受理されるなら、またそのときに限って取り出すようなオートマトンである。検索オートマトンはXPath式から生成する。

【0021】

また、本実施の形態において、スキーマとは、

非終端記号 $x ::=$ 正規表現 r

という形をした規則がいくつか並んだものとする。正規表現 r を構成するシンボルは、タグ名と非終端記号 x の対または属性名と非終端記号 x の対である。いくつかの非終端記号 x は、開始記号として指定されている。スキーマを表現するための言語として、DTD (Document Type Definition)、W3C XML Schema、RELAX (REGular Language description for XML) NGなどが知られている。これらのスキーマ言語で書かれたスキーマを、上記の形のスキーマに変換することは容易

である。したがって本実施の形態は、これらのスキーマ言語で書かれたスキーマにも適用されるものである。

上述のように、本実施の形態では、アクセス制御オートマトンおよび検索オートマトンとスキーマを照合するため、スキーマオートマトンを導入する。スキーマオートマトンとは、スキーマに照らして妥当な文書のルートノードから任意のノードからまでのパスを全て受理し、それ以外のパスは受理しないオートマトンである。

【0022】

図1は、本実施の形態によるアクセス権解析器を実現するのに好適なコンピュータ装置のハードウェア構成の例を模式的に示した図である。

図1に示すコンピュータ装置は、演算手段であるCPU (Central Processing Unit: 中央処理装置) 101と、M/B (マザーボード) チップセット102及びCPUバスを介してCPU 101に接続されたメインメモリ103と、同じくM/Bチップセット102及びAGP (Accelerated Graphics Port) を介してCPU 101に接続されたビデオカード104と、PCI (Peripheral Component Interconnect) バスを介してM/Bチップセット102に接続されたハードディスク105、ネットワークインターフェイス106及びUSBポート107と、さらにこのPCIバスからブリッジ回路108及びISA (Industry Standard Architecture) バスなどの低速なバスを介してM/Bチップセット102に接続されたフロッピーディスクドライブ109及びキーボード/マウス110とを備える。

なお、図1は本実施の形態を実現するコンピュータ装置のハードウェア構成を例示するに過ぎず、本実施の形態を適用可能であれば、他の種々の構成を取ることができる。例えば、ビデオカード104を設ける代わりに、ビデオメモリのみを搭載し、CPU 101にてイメージデータを処理する構成としても良いし、ATA (AT Attachment) などのインターフェイスを介してCD-ROM (Compact Disc Read Only Memory) やDVD-ROM (Digital Versatile Disc Read Only Memory) のドライブを設けても良い。

【0023】

図 2 は、本実施の形態によるアクセス権解析器の機能構成を示すブロック図である。

図 2 に示すように、本実施の形態によるアクセス権解析器 2 0 0 は、パス式から検索オートマトンを構築する検索オートマトン生成部 2 1 0 と、アクセス制御ポリシーからアクセス制御オートマトンを生成するアクセス制御オートマトン生成部 2 2 0 と、スキーマからスキーマオートマトンを生成するスキーマオートマトン生成部 2 3 0 と、これらのオートマトンを用いてアクセス権の判定を行う論理演算部 2 4 0 とを備える。そして、XML データベースの前段で XML 文書の検索前に予備的なアクセス権解析を実行する。

【 0 0 2 4 】

上述した各構成要素は、例えば、図 1 に示したメインメモリ 1 0 3 に展開されたプログラムにて CPU 1 0 1 を制御することにより実現される仮想的なソフトウェアブロックである。CPU 1 0 1 を制御してこれらの機能を実現させるプログラムは、磁気ディスクや光ディスク、半導体メモリ、その他の記憶媒体に格納して配布したり、ネットワークを介して配信したりすることにより提供される。本実施の形態では、図 1 に示したネットワークインターフェイス 1 0 6 やフロッピーディスクドライブ 1 0 9、図示しない CD-ROM ドライブなどを介して当該プログラムを入力し、ハードディスク 1 0 5 に格納する。そして、ハードディスク 1 0 5 に格納されたプログラムをメインメモリ 1 0 3 に読み込んで展開し、CPU 1 0 1 にて実行することにより、これらの機能を実現する。

【 0 0 2 5 】

上記の構成において、検索オートマトン生成部 2 1 0 は、検索条件を指定するパス式を入力して検索オートマトンを生成する。具体的には、解析すべき XPath 式が与えられると、これを正規パス式とみなし、さらに検索オートマトンに変換したものを検索オートマトン q で表現する。生成された検索オートマトンは、例えば図 1 に示したメインメモリ 1 0 3 の作業領域に保持され、論理演算部 2 4 0 にて使用される。

ここで、本実施形態では、XPath を正規パス式 (regular path expression = 正規表現によって所定のノードに至るパスを表した制約式) とみなす手法を

用いる。すなわち、

- ・「/」は、正規表現 ϵ とみなす。ここで、 ϵ は空文字をあらわす。
- ・「//title」は、正規表現 $\Sigma^*.title$ とみなす。ここで、 Σ^* は、任意の要素名の任意の回数の繰り返しをあらわす。また「.」は連結を示し、 $\Sigma^*.title$ とは、任意の要素名の繰り返しの後に `title` がくるような XML 文書のパスを意味する。
- ・「/bib/book/title」は、正規表現 `bib.book.title` とみなす。これは、`bib`、`book`、`title` をこの順で通るようなパスを意味する。

所定のノード（要素，属性，テキスト） n に至るパスがパス正規表現 r に属するならば、ノード n はパス正規表現 r に選択されるという。

【0026】

図3は検索対象となるXML文書の例を示す図である。

図3のXML文書に対する検索を考えると、検索したいXPath式が全ての`author`要素、すなわち`//author`であるならば検索オートマトン q は、図4に示すオートマトンになる。また、XPath式が`bib`要素の下の`book`要素の全ての子供、すなわち`/bib/book/*`であるならば、検索オートマトン q は、図5に示すオートマトンになる。

【0027】

アクセス制御オートマトン生成部220は、アクセス制御のために与えられたアクセス制御ポリシーからアクセス制御オートマトンを生成する。生成されたアクセス制御オートマトンは、例えば図1に示したメインメモリ103の作業領域に保持され、論理演算部240にて使用される。

アクセス制御ポリシーをXACL2（XACLのアクセス制御の伝播を下記のように定義したもの）で記述する場合、そのアクセス制御規則は、サブジェクト（検索の主体）、XPath、アクション（アクセスの内容と可否の情報）の組で表される。

XACL2の各アクセス制御規則が、次のように記述されているとする。

(Admin, /, +r)

(Guest, /bib/book, +r)

(Guest, //price, -r)

一方、XACL2におけるリード・ポリシーの解釈には、次のような原則がある。

- ・読み込み許可+r (grant read +r) は下に伝播する。
- ・読み込み禁止-r (deny read -r) は下に伝播する。
- ・-rは+rに優先し、またデフォルトは-rである。

すなわち、Guestが図3のXML文書をみるとき、/bib/bookの+rが下に伝播し、また//priceの-rが優先であるために、Guestにみえるのはprice要素を除いた全てのノードである。

【0028】

本実施の形態では、サブジェクトが予め与えられているとする。サブジェクトが可変である一般の場合は実施の形態4、5で検討する。

アクセス制御オートマトン生成部220は、上述したようなポリシー記述(+r、-r)とサブジェクト(Admin、Guest)が与えられたとき、次のオートマトン R^+ 、 R^- を計算する。

- ・オートマトン R^+ は、そのサブジェクトに対するアクション+rの許されたXPathに対応するパス正規表現に対応するオートマトン全ての和
- ・オートマトン R^- は、そのサブジェクトに対するアクション-rの許されたXPathに対応するパス正規表現に対応するオートマトン全ての和

正規表現からオートマトンへの変換およびオートマトンに対する集合演算のアルゴリズムは周知である(例えば、非特許文献3参照)。さらに以下のようなアクセス制御オートマトン Γ を計算する。

$$\Gamma = R^+ \cdot \Sigma^* - R^- \cdot \Sigma^*$$

すなわち、アクセス制御オートマトン Γ は、オートマトン R^+ の後に任意のパ

ス Σ^* を連結した正規集合から、オートマトン R^- の後に任意のパス Σ^* を連結した正規集合を取り除いた差分の正規集合を表現するオートマトンである。

【0029】

例を挙げると、上記のゲストに対するオートマトン R^+ は、 $\varepsilon \mid \text{bib. book}$ という正規表現で表すことができ、オートマトン R^- は、 $\Sigma^*. \text{price}$ で表すことができるので、アクセス制御オートマトン Γ は、 $(\varepsilon \mid \text{bib. book}). (\Sigma - \{\text{price}\})^*$ という正規表現で示すことができる。

図6は、かかるアクセス制御オートマトン Γ を説明する図である。

図6に示すアクセス制御オートマトン Γ において、XML文書のインスタンスに対するアクセス判定 (access decision) は、次のようにする。

XML文書において、読み込みアクセスが許可されたノード (要素、属性、テキスト等) の集合 $GRANT$ 、読み込みアクセス権がないノードの集合 $DENY$ を次のように求めることができる。

$n \in GRANT \Leftrightarrow$ ノード n は、アクセス制御オートマトン Γ で選択される。

$DENY =$ 「与えられた文書中の全てのノードの集合」 $- GRANT$

ただし、所定のノード (要素、属性、テキスト) n に至るパスがオートマトン Γ に受理されるならば、ノード n はアクセス制御オートマトン Γ に選択されると言う。

【0030】

スキーマオートマトン生成部230は、スキーマの記述からスキーマオートマトンを生成する。生成されたスキーマオートマトンは、例えば図1に示したメインメモリ103の作業領域に保持され、論理演算部240にて使用される。XML文書のためのスキーマとしては、DTD、XML Schema、RELAX NG等がよく用いられている。本実施の形態では、DTDを用いるものとして説明するが、他の種類のスキーマでも概ね同様である。

次のようなDTDを考える。

```

<!ELEMENT bib (book*)>
<!ELEMENT book (title, author*, price)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT price (#PCDATA)>

```

【 0 0 3 1 】

本実施の形態では、この DTD で許されている任意のパスに対応するパス正規表現に対応するオートマトン S を考える。この DTD で許されるどの XML 文書のどのノードに対するルートからのパスも、このオートマトン S によって受理され、それ以外のパスはオートマトン S によって受理されない。このオートマトン S がスキーマオートマトンである。

DTD を与えられてスキーマオートマトン S を求めるアルゴリズムを以下に示す。

まず、スキーマオートマトン S の状態遷移を求める。各要素名 n に対する状態を q_n とする。いま、`<!ELEMENT n (...)>` のような要素宣言 (element declaration) があった場合、`(...)` の中に要素名 m が出現するとき、要素名 n と状態 q_n とを与えられて、 q_m に遷移する遷移をスキーマオートマトン S に追加する。これを全ての要素宣言に対して行えば、スキーマオートマトン S の状態遷移 (state transition) が求まる。スキーマオートマトン S の初期状態 (initial state) は、ルートの要素名に対応する状態である。また、スキーマオートマトン S の終了状態集合 (final state set) は、全ての状態である。

例えば上記の DTD からは、`bib.(ε|book.(ε|title|author|price))` という正規表現に対応するスキーマオートマトン S が得られる。

【 0 0 3 2 】

論理演算部 2 4 0 は、上記の検索オートマトン生成部 2 1 0、アクセス制御オートマトン生成部 2 2 0 及びスキーマオートマトン生成部 2 3 0 により生成された各オートマトンを用いて、与えられたパス式による XML 文書の所望のノードへのアクセス可否を判定する。ここで判定しようとする内容は、次の 2 つの性質

である。

- ・常に拒否 (always denied) : 検索オートマトン q で検索される全てのノードへのアクセスが拒否されるか否か。

- ・常に許可 (always permitted) : 検索オートマトン q で検索される全てのノードへのアクセスが許可されるか否か。

「常に許可」の判定にも「常に拒否」の判定にも失敗した場合、論理演算部 240 は、「不確定 (indeterminate)」という結果を出力する。

【0033】

「常に許可」と「常に拒否」を判定するためには、次のことを調べればよい（ただし、 q は検索オートマトン、 Γ はアクセス制御オートマトン、 S はスキーマオートマトン）。

「常に拒否」 q が $\Gamma \cap S$ と disjoint

「常に許可」 $q \cap S$ が Γ に含まれる

ここで、 $X \cap Y$ とは、 X と Y の積集合を示す。 X と Y とが disjoint であるとは、 $X \cap Y$ が空集合であることを意味する。含まれるとは、集合の包含 \subseteq を示す。これらのオートマトンに対する演算は、全て、例えば非特許文献 3 に開示された既存の手法を用いて計算することができる。

【0034】

上述した $XPath$ 式 $//author$ に対応する検索オートマトン q と DTD におけるスキーマオートマトン S との共通部分は、 $bib.book.author$ に対応するオートマトンである。これは、上述したアクセス制御ポリシーにおける $Guest$ に対するアクセス制御オートマトン Γ に含まれている。よって、 $XPath$ 式 $//author$ は「常に許可」と判定される。

一方、上述した $XPath$ 式 $bib/book/*$ に対応する検索オートマトン q についてスキーマオートマトン S 、アクセス制御オートマトン Γ を適用すると、次のようになる。

まず、 $q \cap S$ は、 $bib.book.(author|title|price)$ に対応するオートマトンで

ある。アクセス制御オートマトン Γ は、このオートマトンを含まない。次に、 $S \cap \Gamma$ は $\varepsilon | \text{bib.}(\varepsilon | \text{book.}(\varepsilon | \text{title} | \text{author}))$ に対応する、このオートマトンは、 $\text{bib. book.} \Sigma$ のオートマトンと共通部分を持つ。よって、 $XPath式 / \text{bib} / \text{book} / *$ は、「常に許可」でも「常に拒否」でもなく「不確定」である。

【0035】

ここで補足として、上述した判定規則がどうして正しいのかの説明をする。

<「常に拒否」の判定規則>

「 q が $\Gamma \cap S$ と disjoint 」ならば「 q で検索されるノード集合 $\subseteq \text{DENY}$ 」

集合 GRANT について上述した次の性質を使う。

① $n \in \text{GRANT} \Leftrightarrow$ ノード n は Γ で検索される。

いま、ドキュメントの全てのノードに対するパスがスキーマオートマトン S によって受理されるとする。すると、

② ノード n が Γ で検索される \Rightarrow ノード n は $\Gamma \cap S$ によって検索される。

①と②より、 $n \in \text{GRANT}$ ならば、ノード n は $\Gamma \cap S$ によって検索されるということがわかる。したがって、「 q が $\Gamma \cap S$ と disjoint 」であるならば、 q で検索されるノードは $\Gamma \cap S$ によっては検索されないので、 GRANT ではない。

【0036】

したがって、上記の「常に拒否」の規則による判定が成功すれば、検索オートマトン q による検索結果は必ず集合 DENY に含まれることがわかる。

図7は、「常に拒否」の判定規則を説明する図である。

図7において、ノード n_1 は検索オートマトン q で検索されるノード、ノード n_2 、 n_3 はアクセス制御オートマトン Γ で検索されるノードである。すなわち、検索オートマトン q で検索されるノードは、アクセス制御オートマトン Γ で検

索されるノードに含まれないために、必ずDENYである。

【0037】

＜「常に許可」の判定規則＞

「 $q \cap S$ が Γ に含まれる」ならば「 q で検索されるノード集合 $\subseteq \text{GRANT}$ 」

もし、 $q \cap S$ が Γ に含まれるならば、 $n \in q \cap S$ であるノード n はアクセス制御オートマトン Γ に検索されることになる。よって明らかにGRANTである。

したがって、上記の「常に許可」の規則による判定が成功すれば、検索オートマトン q による検索結果は必ず集合GRANTに含まれることがわかる。

ここで、上記の「常に許可」の判定は、アクセス制御オートマトン Γ から上方向に伝播した部分に対する検索が「常に許可」であることを利用することができない。ただし、これを利用する手法を採ることも可能である。

【0038】

このように、本実施の形態によれば、XMLデータベース中のXML文書に対して、データベース検索にかかるパス式のアクセス権が、少なくとも、「常に許可」、「常に拒否」、「不確定」のいずれに該当するかということが、当該XMLデータベースにおいて実際にXML文書の構造を調べて行うアクセス権解析（ノード単位のアクセス権チェック）に先立って判断される。

「常に許可」である場合には、当該パス式を用いたデータベース検索では、XMLデータベースで、XML文書を調べて行う通常のアクセス権解析を行う必要はない。

「常に拒否」である場合は、当該パス式を用いたデータベース検索自体が無効であることが事前にわかる。

したがって、本実施の形態による判定結果が「不確定」である場合にのみ、XML文書を調べて行う通常のアクセス権解析をXMLデータベースにおいて実行すれば良く、データベース検索の処理全体における実行効率を向上させることができる。

【0039】

以上、本実施の形態では、アクセス制御オートマトン Γ とスキーマオートマトン S とを用いて、所定のパス式 q に対しアクセス可否を判定するアクセス権解析

について説明した。ところで、XML文書には、DTDやその他のスキーマが指定されていないものも存在する。そのような場合、スキーマオートマトン S は、 Σ^* （任意の要素名の任意の回数の繰り返し）に対応するオートマトンであるとみなすことにより、上述した本実施の形態のアクセス権解析を適用することができる。

【0040】

〔実施の形態2〕

上記の実施の形態1では、アクセス制御ポリシーからアクセス制御オートマトン Γ を生成し、スキーマからスキーマオートマトン S を生成して、このアクセス制御オートマトン Γ とスキーマオートマトン S とを用いて、所定のパス式 q に対しアクセス可否を判定した。

これに対し、実施の形態2では、DTDなどのスキーマの記述の代わりにパステーブル（path table）を用いてアクセス権解析を行う。

【0041】

ここで、パステーブルとは、XMLデータベースに格納されているXML文書に存在する全てのノード（要素、属性、テキスト等）へのルートからのパスを記憶するデータ構造である。言い換えれば、XMLデータベース中のXML文書における全てのパスを記述した一覧表である。ただし、本実施の形態では、ノードとして要素のみを扱う。

例えば、次のXML文書に対するパステーブルを構築する場合を考える。

```
<foo a0="">
  <bar a-1="">
    <hoge a2="">テキスト</hoge>
  </bar>
  <bar a3=""/>
</foo>
```

このXML文書に含まれる全てのパスを列挙する（ただし、空のテキストを末

端とするパスは省略する) と、次のようになる。

```

／f o o
／f o o／@ a 0
／f o o／b a r
／f o o／b a r／@ a 1
／f o o／b a r／@ a 3
／f o o／b a r／h o g e
／f o o／b a r／h o g e／t e x t ( )
／f o o／b a r／h o g e／@ a 2

```

これらをレコードとしてパステブルが得られる。なお、先頭が@である名前は、属性を表す。

【0042】

本実施の形態によるアクセス権解析器は、上述した実施の形態1と同様に、図1に示したコンピュータ装置等で実現される。

図8は、本実施の形態によるアクセス権解析器の機能構成を示すブロック図である。

図8に示すように、本実施の形態によるアクセス権解析器300は、パス式から検索オートマトンを生成する検索オートマトン生成部210と、アクセス制御ポリシーからアクセス制御オートマトンを生成するアクセス制御オートマトン生成部220と、検索対象であるXML文書のパステブル331を管理するパステブル管理部330と、これらのオートマトンを用いてアクセス権の判定を行うアクセス権判定部としての論理演算部340とを備える。

これらの構成要素のうち、検索オートマトン生成部210及びアクセス制御オートマトン生成部220は、実施の形態1における検索オートマトン生成部210及びアクセス制御オートマトン生成部220と同様であるので、同一の符号を付して説明を省略する。

【0043】

パステブル管理部 330 は、例えば、図 1 に示したプログラム制御された CPU 101 及びメインメモリ 103 やハードディスク 105 等の記憶手段にて実現され、XML データベースに格納されている XML 文書に関して作成された、上述したパステブル 331 を格納して管理する。パステブル 331 は、公知の解析手段を用いて XML データベース中の XML 文書を解析して予め作成しておくが、XML データベースの内容の変更に伴って適宜更新される。

図 9 は、図 3 に示した XML 文書に対して作成されるパステブル 331 の例を示す図である。

図 9 のパステブル 331 には、図 3 の XML 文書における全てのパスが列挙されている。

【0044】

論理演算部 340 は、検索オートマトン生成部 210 及びアクセス制御オートマトン生成部 220 により生成されたオートマトンと、パステブル管理部 330 に格納されているパステブル 331 とを用いて、与えられたパス式による XML 文書の所望のノードへのアクセス可否を判定する。ここで判定しようとする内容は、次の 2 つの性質である。

- ・常に拒否 (always denied) : 検索オートマトン q で検索される全てのノードへのアクセスが拒否されるか否か。
- ・常に許可 (always permitted) : 検索オートマトン q で検索される全てのノードへのアクセスが許可されるか否か。

「常に許可」の判定にも「常に拒否」の判定にも失敗した場合、論理演算部 340 は、「不確定 (indeterminate)」という結果を出力する。

【0045】

XML データベースに対してパステブル 331 が存在する場合、スキーマオートマトンを生成することなく、このパステブル 331 のみを参照してアクセス権解析を行うことができる。例えば、`/bib` という XPath による検索をする場合を考える。図 9 のパステブル 331 が与えられており、アクセス制御ポリシーが XACL 2 にて次のように記述され (なお、`+r` は読み込み許可、`-r` は読み込み禁止)、

(Admin, /, +r)

(Guest, /bib/book, +r)

(Guest, //price, -r)

かつサブジェクトがGuestであったものとする。

【0046】

スキーマオートマトンを用いず、パステブル331のみを用いてアクセス権解析を行うには、例えば、パステブル331の各エントリに対して、アクセス制御においてどう判定されるかを示す情報（以下、判定情報）を予め登録しておく、この情報に従って判定を行うという手法を採ることができる。

図10は、図9のパステブル331にアクセス制御の判定情報を登録した様子を示す図である。

図10に示すパステブル331において、アクセス制御オートマトン Γ で選択されるエントリは、/bib/book、/bib/book/title、/bib/book/authorの3つである。いま、検索オートマトン/bibに対して、これで検索される全てのエントリ（ $=q \cap S$ ）に読み込み許可を示す+rが付加されていたならば、判定結果は「常に許可」である。

反対に、検索オートマトンで検索される全てのパステブルエントリに読み込み禁止を示す-rが付加されていたならば、「 q が $\Gamma \cap S$ とdisjoint」であるという判定が成功したことになるため、判定結果は「常に拒否」である。

検索オートマトンで検索される全てのパステブルエントリに、読み込み許可を示す+rと読み込み禁止を示す-rとが混在しているならば、判定結果は「不確定」となる。

【0047】

さて、上記のアクセス権解析の手法では、スキーマオートマトンを生成せずに、パステブル331の情報のみからアクセスの可否を判定した。これに対し、パステブル331にかかる情報を登録しておくのではなく、パステブル331のエントリからスキーマオートマトンを生成してアクセス権解析を行うことも

可能である。

すなわち、XMLデータベース中のXML文書に対し、予めその内容から図9に示したようなパステブル331が作られていれば、全てのXML文書は、このパステブル331に存在するパスに制約されるので、このパステブル331をスキーマオートマトンとして用いることができる。

【0048】

この場合、図8に示した本実施の形態のアクセス権解析器300に、さらにスキーマオートマトン生成部が設けられ、パステブル管理部330からパステブル331を読み込んでスキーマオートマトンを生成し、論理演算部340に送ることとなる。

パステブル331からスキーマオートマトンを生成するためには、パステブル331の各パスにそれぞれ対応するオートマトンを作り、その和を求めるという手法を採ることができる。すなわち、`/bib/book/title`であれば、`bib.book.title`という正規表現に対応するオートマトンを作り、各オートマトンの和は既知の手法で求めれば良い。

【0049】

このように、本実施の形態によれば、実施の形態1の場合と同様に、XMLデータベース中のXML文書に対して、データベース検索にかかるパス式のアクセス権が、少なくとも、「常に許可」、「常に拒否」、「不確定」のいずれに該当するかということが、当該XMLデータベースにおいて実際にXML文書の構造を調べて行うアクセス権解析（ノード単位のアクセス権チェック）に先立って判断される。

したがって、本実施の形態による判定結果が「不確定」である場合にのみ、XML文書を調べて行う通常のアクセス権解析をXMLデータベースにおいて実行すれば良く、データベース検索の処理全体における実行効率を向上させることができる。

【0050】

〔実施の形態3〕

上記の実施の形態1、2では、検索条件の記述としてXPath等のパス式を

用いたアクセス権解析について説明したが、実施の形態 3 では、パス式を内包する問い合わせ式を用いたアクセス権解析を説明する。

ここで、問い合わせ式 (query expression) とは、XML データベースへの検索を指定する式である。パス式によってノードを見つけ、それに対して何らかの操作 (要素をいくつか追加して新たな XML 文書を構築する操作など) を行う。W3C の XQuery は、問い合わせ式である。XQuery は、パス式の表現に XPath を用いる。本実施の形態では、問い合わせ式として XQuery を用いた例について説明する。

【0051】

本実施の形態によるアクセス権解析器は、上述した実施の形態 1 と同様に、図 1 に示したコンピュータ装置等で実現される。

図 11 は、本実施の形態によるアクセス権解析器の機能構成を示すブロック図である。

図 11 に示すように、本実施の形態によるアクセス権解析器 400 は、パス式から検索オートマトンを構築する検索オートマトン生成部 210 と、アクセス制御ポリシーからアクセス制御オートマトンを生成するアクセス制御オートマトン生成部 220 と、スキーマからスキーマオートマトンを生成するスキーマオートマトン生成部 230 と、これらのオートマトンを用いて個々のパス式に対するアクセス権の判定を行う論理演算部 240 と、問い合わせ式からパス式を抽出するパス式抽出部 450 と、論理演算部 240 による判定結果に基づいて問い合わせ式自体のアクセス権の判定を行う問い合わせ式書き換え・判定部 460 とを備える。

これらの構成要素のうち、検索オートマトン生成部 210、アクセス制御オートマトン生成部 220、スキーマオートマトン生成部 230 及び論理演算部 240 は、実施の形態 1 における対応する構成要素と同様であるので、同一の符号を付して説明を省略する。

【0052】

パス式抽出部 450 は、XQuery 等の問い合わせ式を解析してパス式を抽出し、得られたパス式を検索オートマトン生成部 210 へ送る。パス式抽出部 4

50としては、例えば、図1に示したプログラム制御されたCPU101にて実現された汎用の構文解析器や、XQuery専用の構文解析器を用いることができる。

図12は、問い合わせ式(XQuery)の例を示す図である。

図12の問い合わせ式の場合、各命令は以下のように解釈され実行される。この手法は既存の公知技術である(例えば、非特許文献4参照)。

- ・1行目、8-9行目は、検索結果の全体を<bib>…</bib>で囲む動作。

- ・2行目のfor文は、http://www.bn.comで指定されたXML文書に対して、/bib/bookで選択される各ノードについて3-7行目の動作を繰り返すという動作。

- ・3行目のwhere文は、for文で選択された各ノードを\$bとしたときに、\$b/publisherのテキスト値が“Addison-Wesley”であり、\$b/@yearの値が1991以上である場合に、4-8行の動作を行うことを示す。

- ・4-7行目は、for文で選択された各ノード\$bに対して、その@year値と、titleだけをコピーしたbook要素を作る動作を示す。

以上の動作における/bib/book、\$b/publisherなどがパス式(XPath)である。

【0053】

図13は、図12の問い合わせ式から抽出されるパス式の一覧を示す図表である。

図13において、2行目のdocument (…) /bib/bookと、6行目のdocument (…) /bib/book/titleとでは、XPathの評価のされ方が異なることに注意する。前者では、book要素の子孫の内容は問われず、各book要素自体をfor文で数えあげられれば良いが、6行目のtitle要素は検索の結果であるために、子孫ノードを含めた全ての内容が問われる。すなわち、/bib/book/priceへのアクセスが拒否されていたとしても、for文の実行には何の影響もないが、/bib/book

k/title/subtitleへのアクセスが禁止されていた場合、return文の実行のためにアクセス制御が必要になる。

このようなXPathの評価のされ方の違いを、図13では、グループAとグループBという分類によって示している。for文に対応するのはグループAであり、return文のように子孫ノードの内容を見る場合にはグループBに分類される。また\$b/publisher="Addison Wesley"のようにあるXPath式の結果のテキスト値が問題となっている場合も、グループBに分類されることになる。

【0054】

各グループA、Bに充当するXPath式の分類は以下の通りである。

グループA：子孫を考慮しなくて良いXPath式。

Ex1：Location designator、forに現れる要素のXPath式。

Ex2：Attribute、様々なところに現れるだろう属性のXPath式。

Ex3：Location designator、様々なところに現れるだろう要素のXPath式（/test[. follows xml]における/test/xmlや/test/）。

Ex4：Location designator、様々なところに現れるだろう要素のXPath式（namespace-uri(. /test)='http://uri'における/test）。

グループB：子孫のノードに対するアクセスの意味を含んだXPath式。

Ex1：returnに現れる要素のXPath式（ある要素のsubtreeなどを考える場合）。

Ex2：（様々なところに現れるだろう）テキスト値を参照する要素のXPath式（/test[. /xml='ABC']の/test/xml）。

【0055】

問い合わせ式から取り出された各パス式qに対して、検索オートマトン生成部

210にて検索オートマトンが生成され、アクセス制御オートマトン生成部220及びスキーマオートマトン生成部230にて生成された各オートマトンと共に、論理演算部240に入力されてアクセス権解析が行われ、以下のようにして、「常に許可」、「常に拒否」、「不確定」の判定が行われる。

- ・「常に拒否」の判定は、グループA、Bの別なくそのままパス式 q に対して適用する。
- ・「常に許可」の判定は、パス式 q がグループAに属するならば、パス式 q に対してそのまま適用し、パス式 q がグループBに属するならば、 $q \cdot \Sigma^*$ に変形してから適用する。
- ・「常に拒否」、「常に許可」のいずれの判定にも失敗したパス式 q は、「不確定」であると判断される。

これらの判定結果は、例えば図1のメインメモリ103の作業領域に保持される。

【0056】

上記のように判定する理由は次の通りである。

「グループAのパス式 q が常に拒否である」とは「パス式 q の検索結果の全てのノードに対するアクセスが拒否である」ということであり、「グループBのパス式 q が常に拒否である」とは、「パス式 q の検索結果の全てのノードに対するアクセスが拒否であり、その子孫ノードもまた見ることはできない」という意味である。しかし、XACLによるアクセス制御ポリシーの記述では、所定のノードに対するアクセスが禁止であれば、その子孫も見ることにはできないのであるから、「常に拒否」に関してはグループAとグループBとで差異はない。

一方、「常に許可」に関しては、「グループAのパス式 q が常に許可である」とは、「パス式 q の検索結果の全てのノードに対するアクセスが許可される」ということである。これに対し、「グループBのパス式 q が常に許可である」とは、「パス式 q の検索結果の全てのノードに対するアクセスが許可され、その子孫ノードもまた全て見ることはできる」という意味となり、両者は異なる。グルー

ブBの場合は、全ての子孫ノードへのアクセスの許可を保障するため、 q, Σ^* で検索される全てのノードに対して、アクセス権の判定が成功しなければならない。したがって、上述したような判定手法を行うことが必要である。

図14は、図12の問い合わせ式から抽出された個々のパス式に対するアクセス権解析の結果（判定結果）を示す図表である。

【0057】

問い合わせ式書き換え・判定部460は、図14のように得られた個々のパス式に対する判定結果から問い合わせ式自体に対するアクセス可否の判定を行う。

問い合わせ式書き換え・判定部460は、まず、「常に拒否」と判断されたパス式に対して空集合を示す値（ここでは\$emptyとする）を代入する。

図15は、図12の問い合わせ式に対してかかる書き換えが行われた状態を示す図である。

この問い合わせ式に残っているパス式は、全て「常に許可」である。一般に、問い合わせ式に含まれる全てのパス式が「常に許可」か「常に拒否」であって「不確定」ではない場合、その問い合わせ式を「常に許可」であるパス式しか含まないように書き換えることができる。これは、「常に拒否」であるパス式については、そもそも当該パス式を用いたデータベース検索自体が無効となるからである。

したがって、このような問い合わせ式は、問い合わせ式書き換え・判定部460によって、「常に許可」とであると判定され、実行時に動的なアクセス制御を行うことなく評価実行できる。問い合わせ式が「不確定」と判定されたパス式を含む場合は、問い合わせ式書き換え・判定部460によって、問い合わせ式自体が「不確定」と判定される。

以上のようにして、問い合わせ式書き換え・判定部460からは、必要に応じて書き換えられた問い合わせ式と、「常に許可」または「不確定」の判定結果とが出力される。

【0058】

図16は、本実施の形態によるアクセス権解析器400の動作を説明するフローチャートである。

図16に示すように、アクセス権解析器400に処理対象である問い合わせ式が入力されると（ステップ1601）、まずパス式抽出部450により問い合わせ式中のパス式が抽出される（ステップ1602）。そして、検索オートマトン生成部210により検索オートマトンが生成され、アクセス制御オートマトン生成部220によりアクセス制御オートマトンが生成され、スキーマオートマトン生成部230によりスキーマオートマトンが生成される（ステップ1603）。

次に、生成された各オートマトンを用いて、論理演算部240により、問い合わせ式から抽出された個々のパス式のアクセス権が判定される（ステップ1604）。

【0059】

次に、この個々のパス式のアクセス権判定結果に基づいて、問い合わせ式書き換え・判定部460により問い合わせ式のアクセス権が判定される。

まず、「常に拒否」と判定されたパス式がある場合は、そのパス式が空集合を示す値に書き換えられる（ステップ1605、1606）。その後、全てのパス式に対する判定結果に基づいて、問い合わせ式が「常に許可」または「不確定」と判定され、この判定結果が書き換えられた問い合わせ式と共に出力される（ステップ1607）。出力された判定結果及び問い合わせ式は、XMLデータベースに渡され、判定結果に応じてアクセス権解析が行われた後、もしくは直ちに当該問い合わせ式によるデータベース検索が行われる。

【0060】

このように、本実施の形態によれば、XMLデータベース中のXML文書に対して、データベース検索にかかる問い合わせ式のアクセス権が、少なくとも、「常に許可」、「不確定」のいずれに該当するかということが、当該XMLデータベースにおいて実際にXML文書の構造を調べて行うアクセス権解析（ノード単位のアクセス権チェック）に先立って判断される。

「常に許可」である場合には、当該問い合わせ式によるデータベース検索では、XMLデータベースで、XML文書を調べて行う通常のアクセス権解析を行う必要はない。

したがって、本実施の形態による判定結果が「不確定」である場合にのみ、X

ML文書を調べて行う通常のアクセス権解析をXMLデータベースにおいて実行すれば良く、データベース検索の処理全体における実行効率を向上させることができる。

【0061】

なお、上記のように問い合わせ式を書き換えずに、その解釈実行において各パス式に対する「常に許可」、「常に拒否」の判定を利用することもできる。この場合、問い合わせ式の解釈実行システムは、パス式に対するアクセス権解析器の評価結果を組み合わせて実現されているものとする。このアクセス権解析器は、「常に許可」というパス式に対する検索結果を取り出すときには、データベースに対する実行時アクセス制御を行わず、また「常に拒否」というパス式に対しては、検索自体を行わないですむ。

【0062】

本実施の形態では、アクセス権解析器400は、実施の形態1のアクセス権解析器200にパス式抽出部450及び問い合わせ式書き換え・判定部460を付加した構成について説明したが、スキーマオートマトンの生成を行わない実施の形態2のアクセス権解析器300にパス式抽出部450及び問い合わせ式書き換え・判定部460を付加した構成とすることもできるのは言うまでもない。この場合、アクセス権解析器400は、図8に示したように検索オートマトン生成部210、アクセス制御オートマトン生成部220、パステーブル管理部330及び論理演算部340を備え、さらにパス式抽出部450と問い合わせ式書き換え・判定部460とを備える。

【0063】

アクセス権解析器400の動作においても、上記ステップ1603において、スキーマオートマトンの生成は行われぬ。またステップ1604において、スキーマオートマトンの代わりにパステーブルを用いて、個々のパス式に対して「常に許可」、「常に拒否」、「不確定」の判定が行われるが、この動作については実施の形態2と同様である。その他の動作については上記と同様である。

【0064】

〔実施の形態4〕

実施の形態 4、5 では、実施の形態 1～3 で説明したアクセス権解析器 2 0 0、3 0 0、4 0 0 の実装例について説明する。実施の形態 1～3 では、アクセス制御を受けるサブジェクト (s u b j e c t) が与えられた場合の静的な判定手法を説明したが、実際の実行環境では、アクセス制御のサブジェクトは動的に決定される。そこで、サブジェクトの特定を含め、データベース検索システムに実装されたアクセス権解析器について説明する。

問い合わせ式に対するアクセス権解析を実行するタイミングとしては、データベース検索の「実行直前」に行うシステムと、問い合わせ式の「コンパイル時」に行うシステムとが考えられる。実施の形態 4 では、「実行直前」に行うシステムを説明する。

【 0 0 6 5 】

図 1 7 は、アクセス権解析を実行時に行うデータベース検索システムの構成例を示す図である。

図 1 7 を参照すると、本実施の形態によるデータベース検索システムは、検索対象である XML 文書を格納した XML データベース 1 0 と、検索条件を示す問い合わせ式 (X Q u e r y) を入力してパス式 (X P a t h) を抽出するパス式抽出部 2 0 と、パス式抽出部 2 0 にて抽出されたパス式を用いて XML データベース 1 0 へのアクセス権限を判断するアクセス権解析器 3 0 と、アクセス権解析器 3 0 の解析結果に応じて問い合わせ式を書き換えると共に、当該問い合わせ式のアクセス権を判定する問い合わせ式書き換え・判定部 4 0 とを備える。

このアクセス権解析器 3 0 として、実施の形態 1、2 で説明したアクセス権解析器 2 0 0、3 0 0 を用いることができる。また、パス式抽出部 2 0 は実施の形態 3 で説明したパス式抽出部 4 5 0 と同一であり、問い合わせ式書き換え・判定部 4 0 は実施の形態 3 で説明した問い合わせ式書き換え・判定部 4 6 0 と同一である。したがって、アクセス権解析器 3 0 と、パス式抽出部 2 0 及び問い合わせ式書き換え・判定部 4 0 とを合わせると、実施の形態 3 で示したアクセス権解析器 4 0 0 の構成と等しい。

【 0 0 6 6 】

さて、データベース検索の実行直前にアクセス権解析を行う本実施の形態のシ

ステムの場合、以下のような特徴がある。

- ・ パステーブルがある場合、最新の内容を参照することができる。
- ・ サブジェクトが決定している。

したがって、パステーブルを用いてパス式のアクセス可否の判定を行う、実施の形態 2 で説明したアクセス権解析器 3 0 0 を、本実施の形態のアクセス権解析器 3 0 として用いることが好ましい。

【 0 0 6 7 】

図 1 7 に示したデータベース検索システムにおいて、所定の問い合わせ式がパス式抽出部 2 0 に入力されると、この問い合わせ式からパス式が抽出されて、アクセス権解析器 3 0 に送られる。このとき、問い合わせ式による検索の主体であるサブジェクトも特定される。なお、問い合わせ式からパス式を抽出する処理は、問い合わせ式のコンパイル時に 1 回だけ行えば良い。そして、問い合わせ式と抽出されたパス式群を対応付けたテーブル（例えば図 1 3 や図 1 4 に示したテーブル）等を作成して管理しておくことにより、データベース検索を実行する度にパス式を抽出する煩雑さを避けられる。

アクセス権解析器 3 0 で解析が行われた後、問い合わせ式書き換え・判定部 4 0 によって、「常に拒否」と判断されたパス式の問い合わせ式における記述が書き換えられる。そして、「常に許可」もしくは「不確定」となった問い合わせ式が XML データベース 1 0 へ送られる。

XML データベース 1 0 では、問い合わせ式書き換え・判定部 4 0 での判定結果が「常に許可」である問い合わせ式に関しては、XML 文書の構造を調べて行う通常のアクセス権解析を行うことなく、直ちに XML 文書の検索を実行する。また、問い合わせ式書き換え・判定部 4 0 での判定結果が「不確定」である問い合わせ式に関しては、XML 文書の構造を調べて行う通常のアクセス権解析を行った後に、XML 文書の検索を実行する。

【 0 0 6 8 】

本実施の形態によるデータベース検索システムの場合、アクセス権解析をデー

データベース検索の実行時に行うため、問い合わせ式を発行してから検索結果を得るまでの処理にその分だけ時間を要することとなる。しかし、各パス式とサブジェクトとの組に対するアクセス権解析の結果をキャッシュしておくことによって、処理性能を向上させる（処理に要する時間を短縮する）ことができる。ただし、このキャッシュはパステーブルが更新されると無効になるので、そのたびにリセットすることが必要である。

【 0 0 6 9 】

〔実施の形態 5〕

実施の形態 5 では、問い合わせ式の「コンパイル時」にアクセス権解析を行うシステムを説明する。

図 1 8 は、アクセス権解析を実行時に行うデータベース検索システムの構成例を示す図である。

図 1 8 を参照すると、本実施の形態によるデータベース検索システムは、検索対象である XML 文書を格納した XML データベース 1 0 と、検索条件を示す問い合わせ式（X Q u e r y）を入力してパス式（X P a t h）を抽出するパス式抽出部 2 0 と、パス式抽出部 2 0 にて抽出されたパス式を用いて XML データベース 1 0 へのアクセス権限を判断するアクセス権解析器 3 0 と、アクセス権解析器 3 0 の解析結果に応じて問い合わせ式を書き換えると共に、当該問い合わせ式のアクセス権を判定する問い合わせ式書き換え・判定部 4 0 と、問い合わせ式書き換え・判定部 4 0 による判定結果を保持する記憶部 5 0 とを備える。

このアクセス権解析器 3 0 として、実施の形態 1、2 で説明したアクセス権解析器 2 0 0、3 0 0 を用いることができる。また、アクセス権解析器 3 0 と、パス式抽出部 2 0 及び問い合わせ式書き換え・判定部 4 0 とを合わせると、実施の形態 3 で示したアクセス権解析器 4 0 0 の構成と等しいことは、実施の形態 4 の場合と同様である。

記憶部 5 0 は、キャッシュメモリであり、例えばデータベース検索システムを図 1 に示したハードウェアで構成した場合、メインメモリ 1 0 3 で実現される。

【 0 0 7 0 】

さて、問い合わせ式のコンパイル時に予めアクセス権解析を行う本実施の形態

のシステムの場合、以下のような特徴がある。

- ・アクセス権解析後にもパステابلが更新され得るため、パステابلを用いることができない。
- ・サブジェクトは決定していない。

したがって、アクセス可否の判定にパステابلを用いない実施の形態1で説明したアクセス権解析器200を、本実施の形態のアクセス権解析器30として用いることが好ましい。

【0071】

図18に示したデータベース検索システムにおいて、所定の問い合わせ式がパス式抽出部20に入力されると、この問い合わせ式からパス式が抽出されて、アクセス権解析器30に送られる。この時点ではサブジェクトは特定されていないため、可能な様々なサブジェクトを想定してアクセス権解析器30によるアクセス権解析が行われる。

アクセス権解析器30で解析が行われた後、問い合わせ式書き換え・判定部40によって、「常に拒否」と判断されたパス式の問い合わせ式における記述が書き換えられる。そして、「常に許可」もしくは「不確定」となった問い合わせ式が記憶部50へ送られ、キャッシュされる。

データベース検索の実行時には、記憶部50にキャッシュされているアクセス権解析結果を用いて問い合わせ式のアクセス可否が認識される。そして、判定結果が「常に許可」である問い合わせ式に関しては、XML文書の構造を調べて行う通常のアクセス権解析を行うことなく、直ちにXML文書の検索を実行する。また、問い合わせ式書き換え・判定部40での判定結果が「不確定」である問い合わせ式に関しては、XML文書の構造を調べて行う通常のアクセス権解析を行った後に、XML文書の検索を実行する。なお、所定のサブジェクトから発行された所定の問い合わせがキャッシュにヒットしない場合には、当該問い合わせ式のコンパイルをやり直す。

【0072】

本実施の形態によるデータベース検索システムの場合、アクセス権解析を問い合わせ式のコンパイル時に行うため、データベース検索の実行時には記憶部 5 0 にキャッシュされた判定結果のみを参照することとなり、高速な処理を実現することが可能である。しかしながら、実施の形態 4 と比較すると、予め可能なサブジェクトを想定して、その全ての場合の判定結果を記憶部 5 0 に保持しておかなければならないため、記憶部 5 0 を実現する記憶手段における記憶容量の使用量が大きい。

【0073】

なお、上述した各実施の形態では、検索オートマトン、アクセス制御オートマトン、スキーマオートマトンにおいて、パスのみを扱っている。パスは単なる列であり、普通のオートマトンによる取り扱いが簡単であるという長所を持つ。しかし、XML 文書の構造は木構造であり、スキーマは XML 文書がどんな木構造を許容するかを表している。アクセス制御ポリシーの記述に用いる X P a t h 検索式も、一般的には木構造に関するものであり、パスだけを扱うものではない。したがって、上記各実施の形態で示した列を扱うオートマトンではなく、木構造を扱うオートマトンを用いて上述したアクセス権解析器 2 0 0、3 0 0、4 0 0、3 0 を実現することも可能である。

【0074】

また、上記各実施の形態では、XML 文書を処理対象として、XML データベースの検索におけるアクセス権解析について説明したが、データベース検索以外の場面でも、広く X P a t h 等のパス式を条件とするアクセス権限の解析を行う場合に、上記各実施の形態を適用できるのは言うまでもない。

さらに、XML 以外の構造化文書を対象とするシステムにおいても、スキーマやアクセス制御ポリシーに相当する仕様が定義されているならば、上記各実施の形態を適用することが可能である。

【0075】

【発明の効果】

以上説明したように、本発明によれば、XML データベースにおいて、XML 文書自体やそのノードを調べることなくアクセス権解析を行うことを実現し、X

MLデータベースの検索性能を向上させることができる。

【図面の簡単な説明】

【図 1】 本実施の形態によるアクセス権解析器を実現するのに好適なコンピュータ装置のハードウェア構成の例を模式的に示した図である。

【図 2】 実施の形態 1 によるアクセス権解析器の機能構成を示すブロック図である。

【図 3】 本実施の形態の検索対象となる XML 文書の例を示す図である。

【図 4】 図 3 の XML 文書に対して、検索したい X P a t h 式が全ての a u t h o r 要素である場合の検索オートマトンを示す図である。

【図 5】 図 3 の XML 文書に対して、検索したい X P a t h 式が b i b 要素の下の子要素の全ての b o o k 要素の子である場合の検索オートマトンを示す図である。

【図 6】 本実施の形態におけるアクセス制御オートマトンを説明する図である。

【図 7】 本実施の形態における「常に拒否」の判定規則を説明する図である。

【図 8】 実施の形態 2 によるアクセス権解析器の機能構成を示すブロック図である。

【図 9】 図 3 に示した XML 文書に対して作成されるパステーブルの例を示す図である。

【図 1 0】 図 9 のパステーブルにアクセス制御の判定情報を登録した様子を示す図である。

【図 1 1】 実施の形態 3 によるアクセス権解析器の機能構成を示すブロック図である。

【図 1 2】 問い合わせ式 (X Q u e r y) の例を示す図である。

【図 1 3】 図 1 2 の問い合わせ式から抽出されるパス式の一覧を示す図表である。

【図 1 4】 図 1 2 の問い合わせ式から抽出された個々のパス式に対するアクセス権解析の結果 (判定結果) を示す図表である。

【図 1 5】 図 1 2 の問い合わせ式に対してかかる書き換えが行われた状態を示す図である。

【図 1 6】 本実施の形態によるアクセス権解析器の動作を説明するフローチャートである。

【図 1 7】 実施の形態 4 によるアクセス権解析を実行時に行うデータベース検索システムの構成例を示す図である。

【図 1 8】 実施の形態 5 によるアクセス権解析を実行時に行うデータベース検索システムの構成例を示すブロック図である。

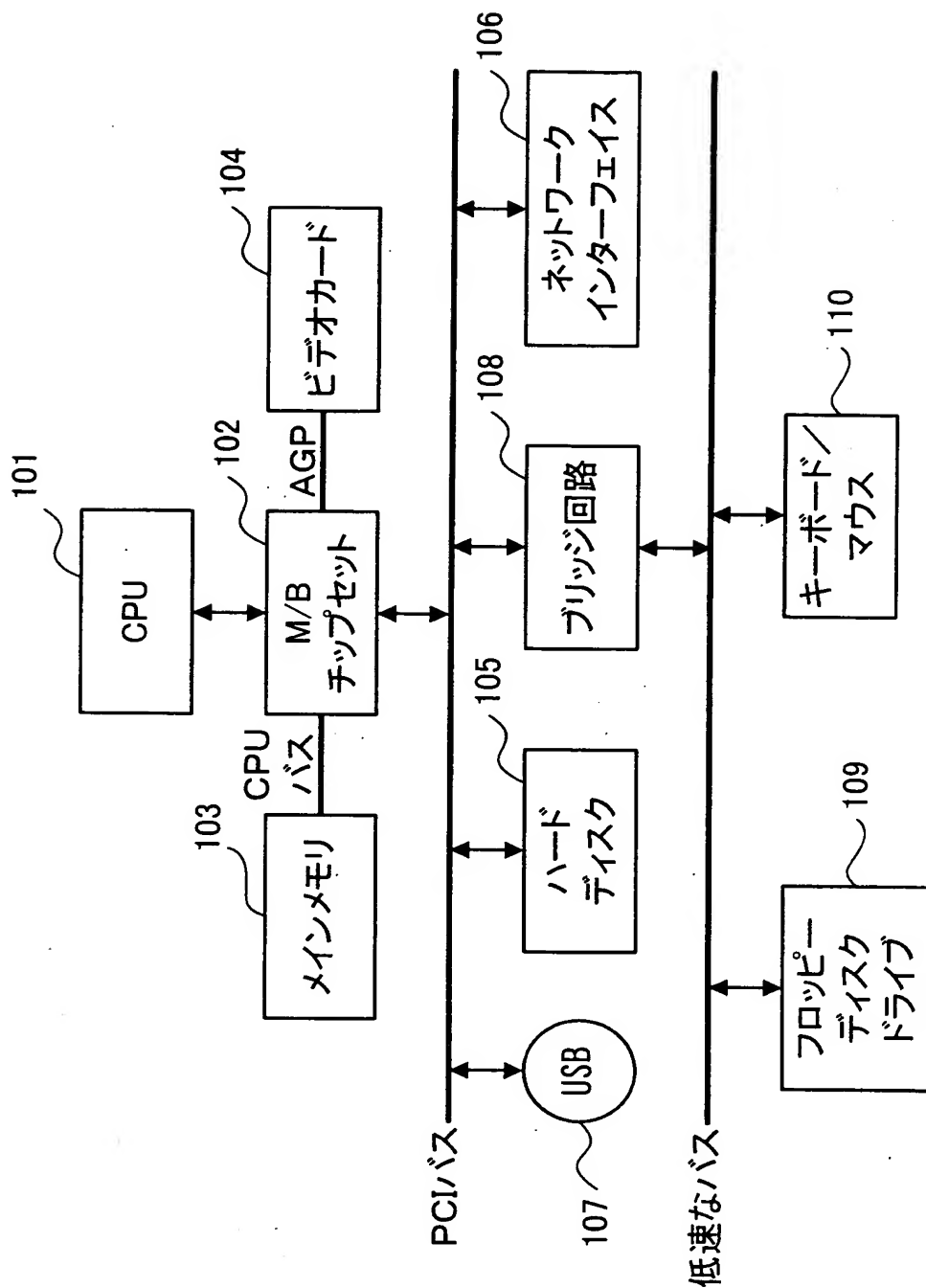
【符号の説明】

1 0 … XML データベース、2 0、4 5 0 … パス式抽出部、3 0、2 0 0、3 0 0、4 0 0 … アクセス権解析器、4 0、4 6 0 … 問い合わせ式書き換え・判定部、2 1 0 … 検索オートマトン生成部、2 2 0 … アクセス制御オートマトン生成部、2 3 0 … スキーマオートマトン生成部、2 4 0、3 4 0 … 論理演算部、3 3 0 … パステーブル管理部、3 3 1 … パステーブル、1 0 1 … CPU、1 0 2 … M/B チップセット、1 0 3 … メインメモリ、1 0 5 … ハードディスク

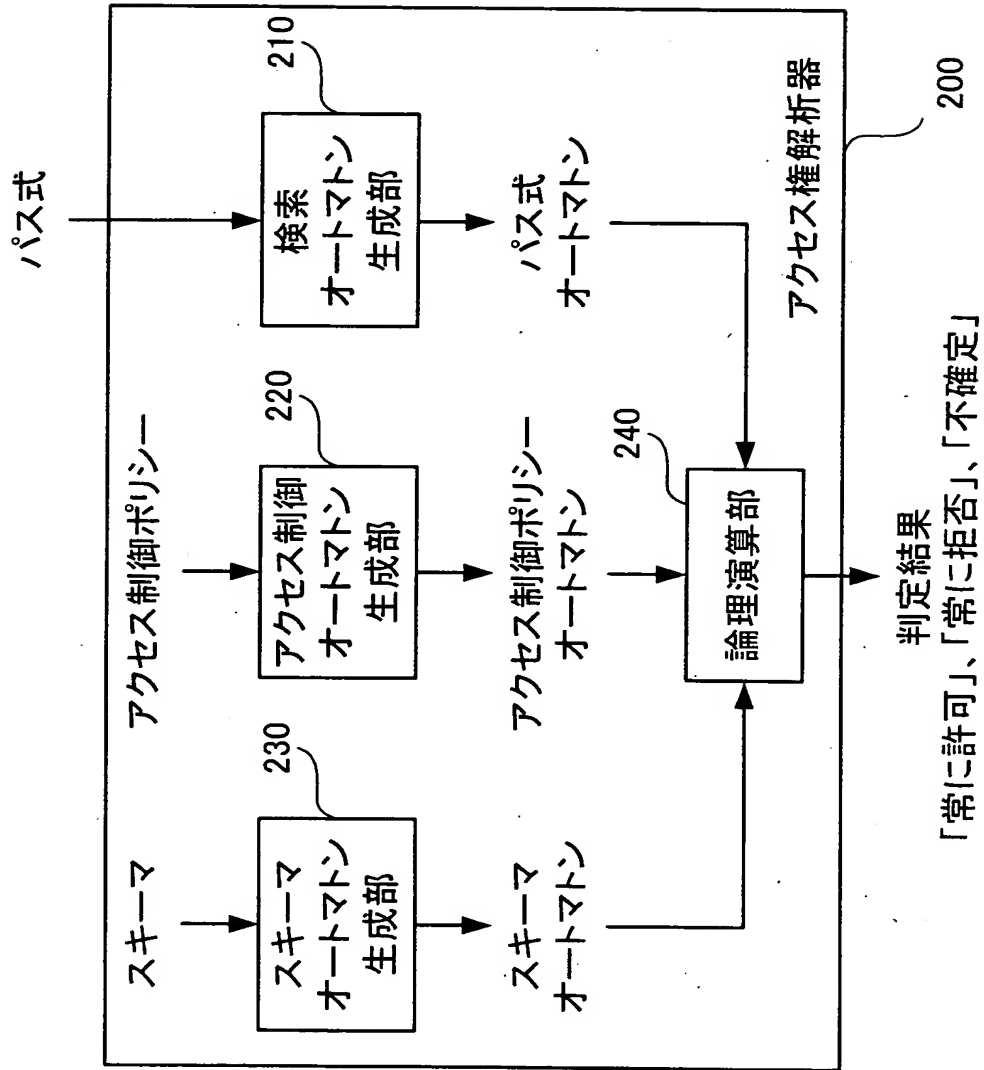
【書類名】

図面

【図 1】



【図 2】



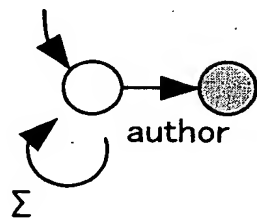
【図 3】

```

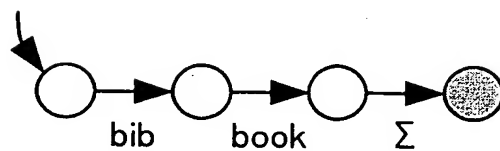
<bib>
  <book>
    <title>TCP/IP illustrated</title>
    <author>W. Stevens</author>
    <price>50. 00</price>
  </book>
  <book>
    <title>Data on the Web</title>
    <author>S. Abiteboul</author>
    <author>P. Buneman</author>
    <author>D. Suciu</author>
    <price>40. 00</price>
  </book>
</bib>

```

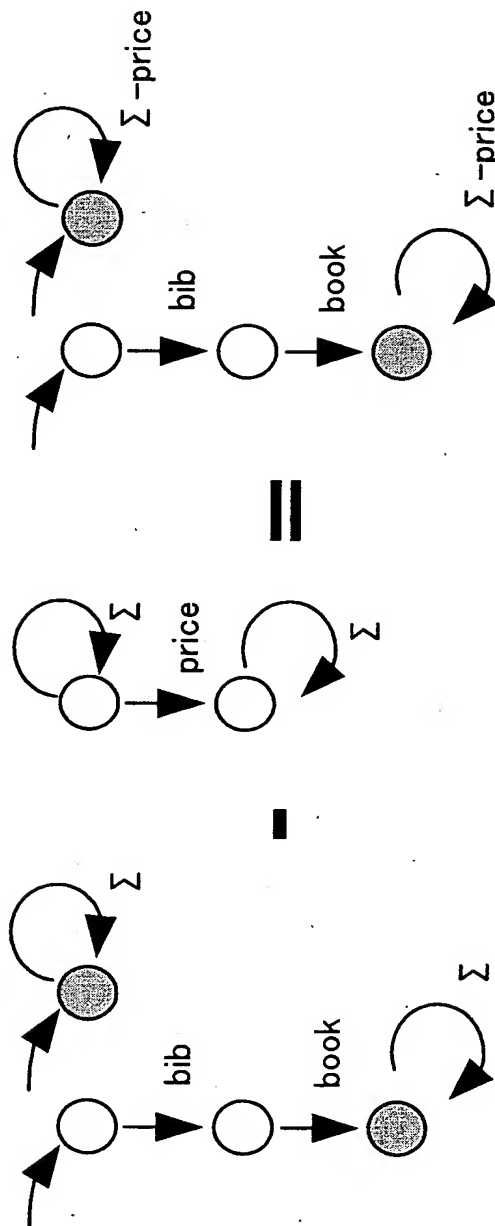
【図 4】



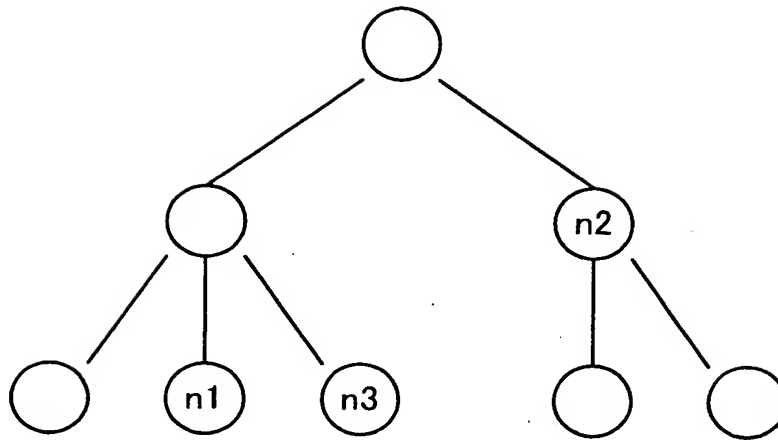
【図 5】



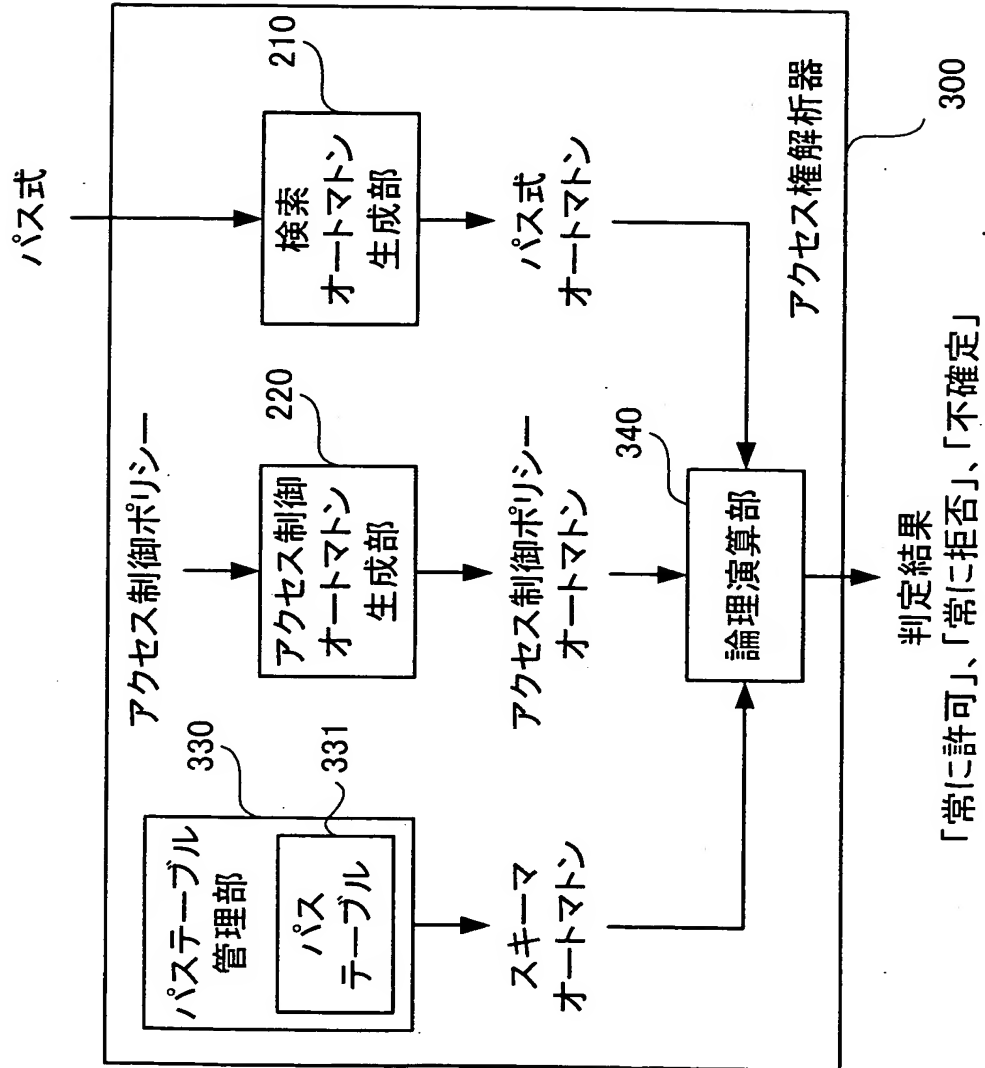
【図 6】



【図 7】



【図 8】



【図 9】

331

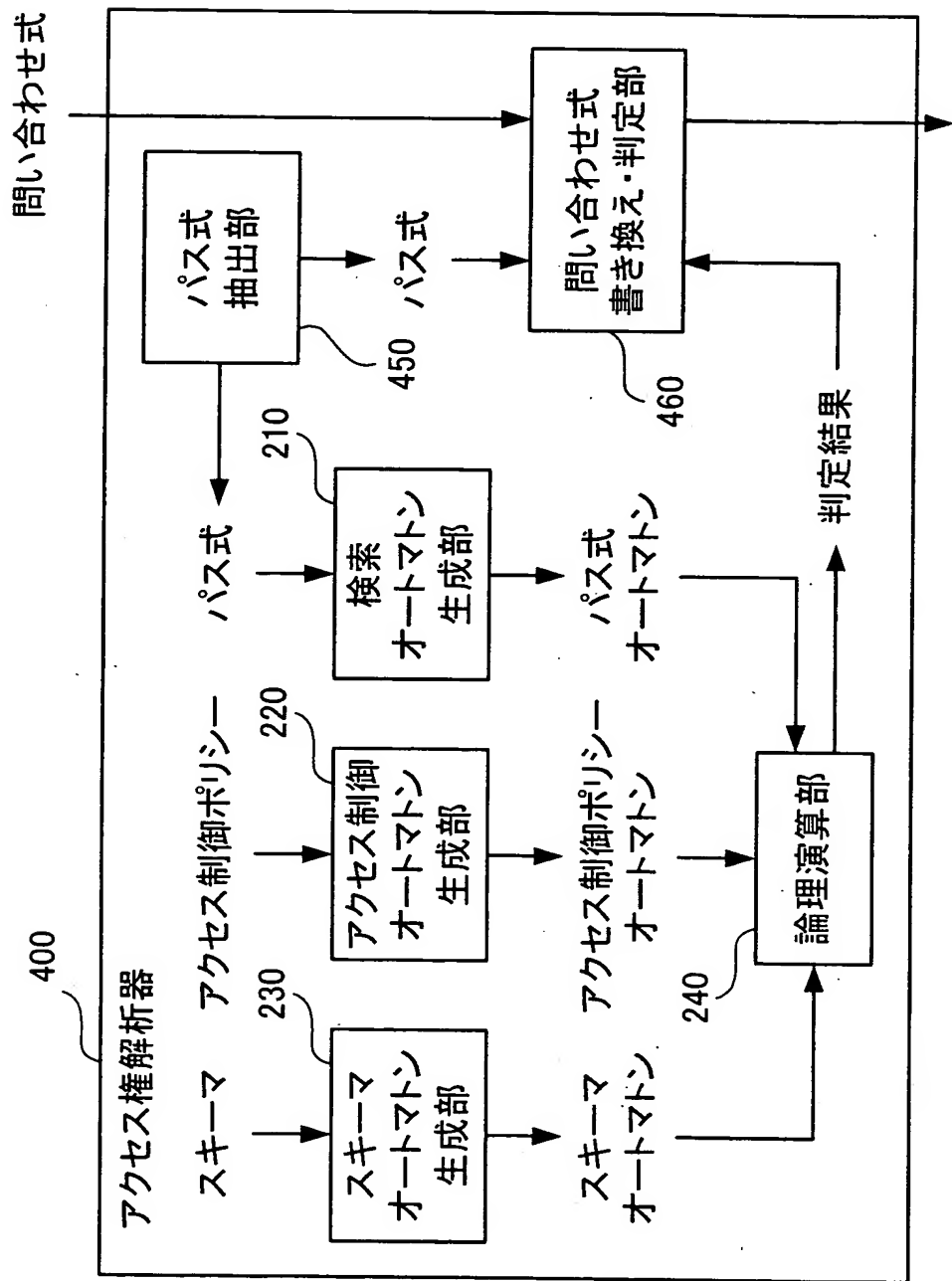
| |
|------------------|
| /bib |
| /bib/book |
| /bib/book/title |
| /bib/book/author |
| /bib/book/price |

【図 1 0】

331

| | |
|------------------|----|
| /bib/book | +r |
| /bib/book/title | +r |
| /bib/book/author | +r |
| /bib/book/price | -r |

【図 11】



書き換えられた問い合わせ式、
判定結果「常に許可」、「不確定」

【図 1 2】

```
1. <bib> {  
2.   for $b in document("http://www.bn.com")/bib/book  
3.   where $b/publisher = "Addison-Wesley" and $b/@year > 1991  
4.   return  
5.     <book year={ $b/@year }>  
6.       { $b/title }  
7.     </book>  
8. }  
9. </bib>
```

【図 1 3】

| 問い合わせ式上の表現 | 取り出されたパス式 (XPath) | グループ |
|-------------------------|--|------|
| 2 行目, document .. /book | document ("http://www. bn. com") /bib/book | A |
| 3 行目, \$b/publisher | document ("http://www. bn. com") /bib/book/publisher | B |
| 3 行目, \$b/@year | document ("http://www. bn. com") /bib/book/@year | B |
| 5 行目, \$b/@year | document ("http://www. bn. com") /bib/book/@year | B |
| 6 行目, \$b/title | document ("http://www. bn. com") /bib/book/title | B |

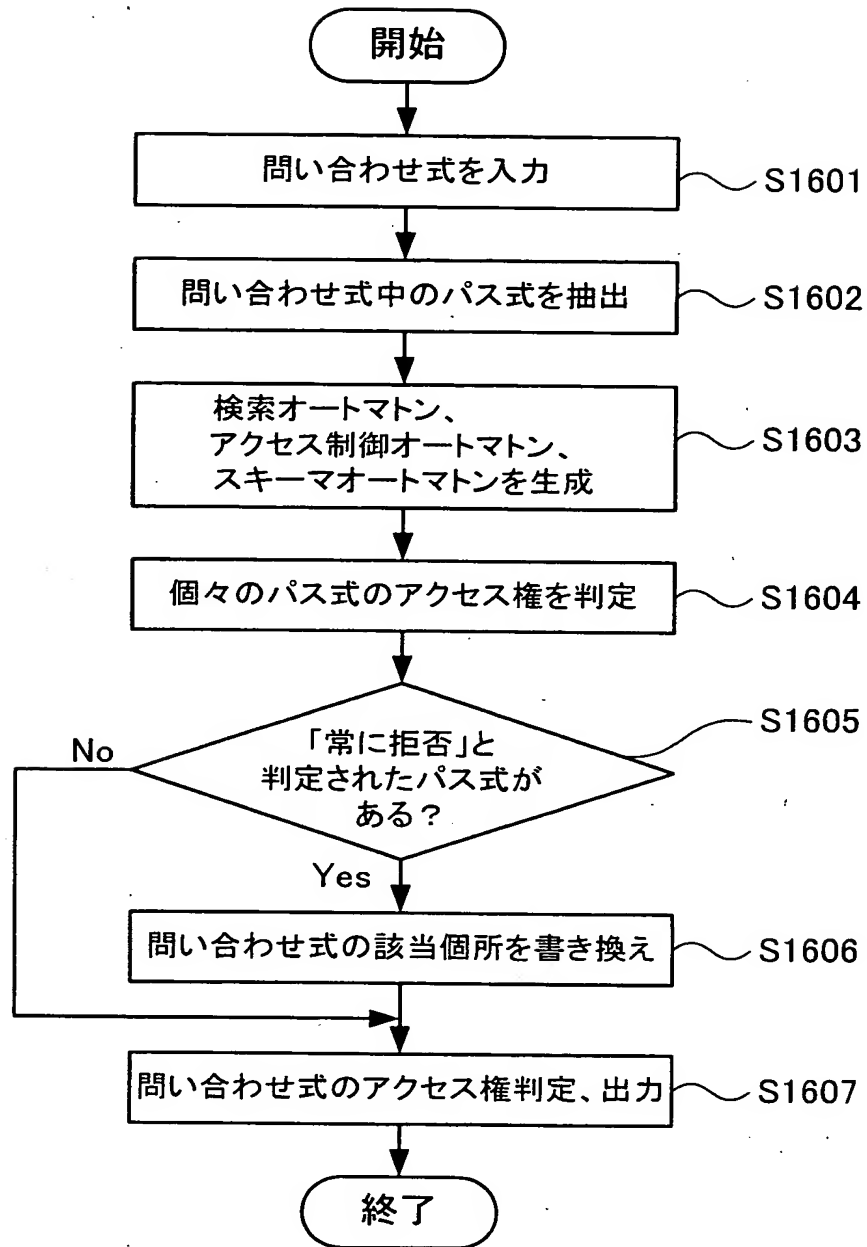
【図 14】

| 問い合わせ式上の表現 | 取り出されたパス式(XPath) | 解析結果 |
|-------------------------|---|------|
| 2 行目, document .. /book | document ("http://www.bn.com")/bib/book | 常に許可 |
| 3 行目, \$b/publisher | document ("http://www.bn.com")/bib/book/publisher | 常に許可 |
| 3 行目, \$b/@year | document ("http://www.bn.com")/bib/book/@year | 常に拒否 |
| 5 行目, \$b/@year | document ("http://www.bn.com")/bib/book/@year | 常に拒否 |
| 6 行目, \$b/title | document ("http://www.bn.com")/bib/book/title | 常に許可 |

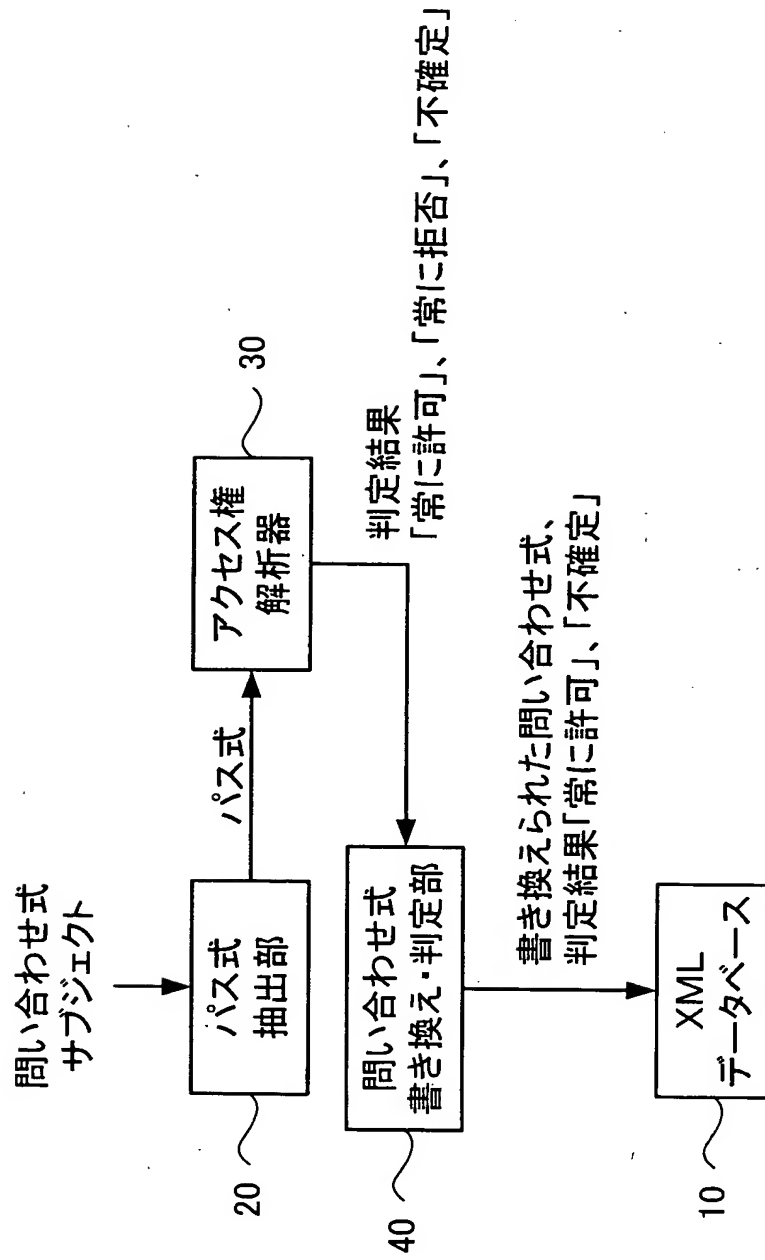
【図 1 5】

```
1. <bib> {  
2.   for $b in document("http://www.bn.com")/bib/book  
3.   where $b/publisher = "Addison-Wesley" and $empty > 1991  
4.   return  
5.     <book year={ $empty }>  
6.       { $b/title }  
7.     </book>  
8. }  
9. </bib>
```

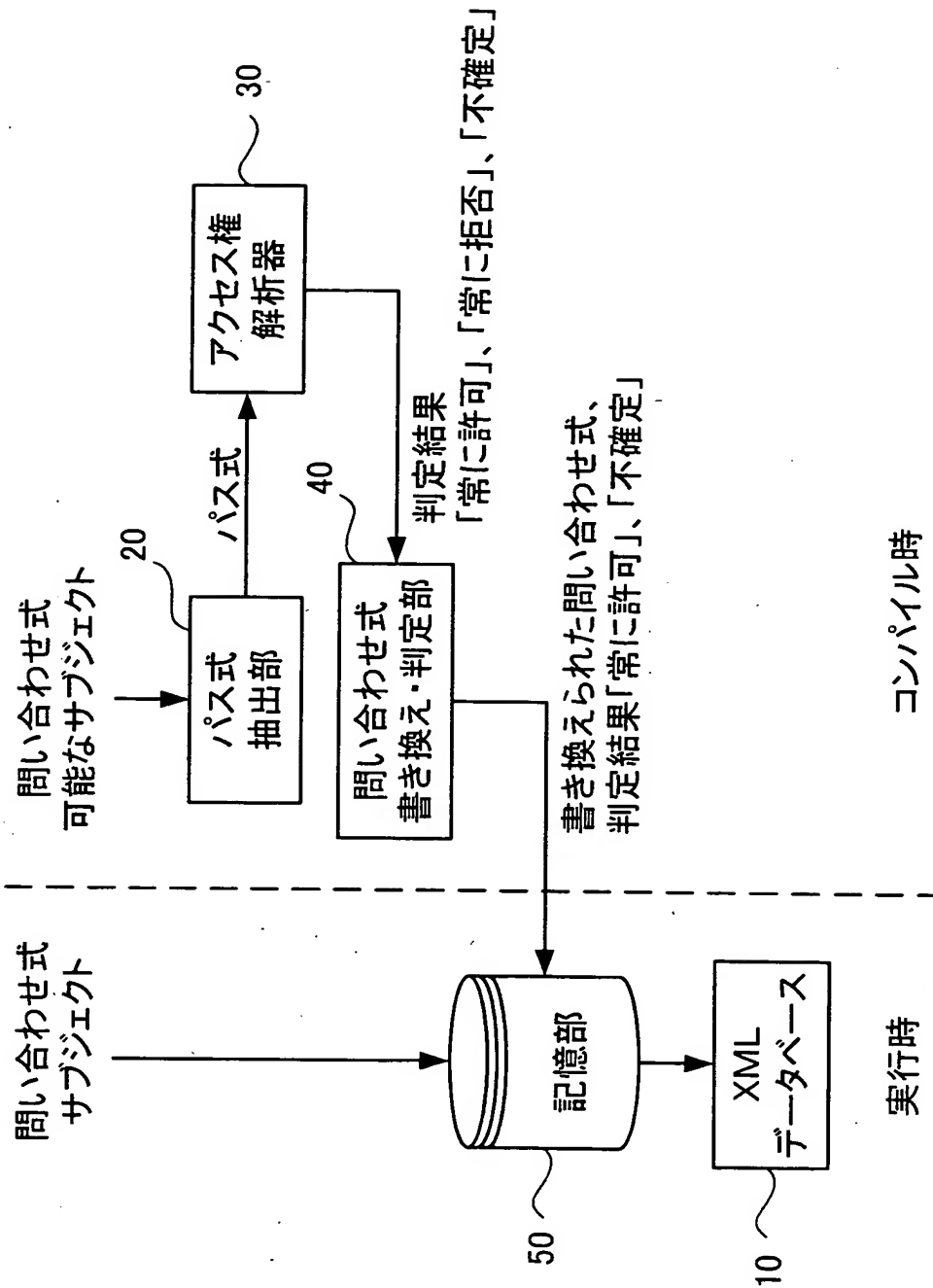
【図 16】



【図 17】



【図18】



【書類名】 要約書

【要約】

【課題】 XMLデータベースにおいて、XML文書自体やそのノードを調べることなくアクセス権解析を行うことを実現し、XMLデータベースの検索性能を向上させる。

【解決手段】 データベースに対する検索条件を記述したパス式から検索オートマトンを生成する検索オートマトン生成部210と、アクセス制御ポリシーからアクセス制御オートマトンを生成するアクセス制御オートマトン生成部220と、スキーマからスキーマオートマトンを生成するスキーマオートマトン生成部230と、生成された各オートマトンに関する論理演算を行う論理演算部240とを備える。これによって、XMLデータベースにおいてXML文書自体を調べ、ノードごとのチェックを行うことなく、かかるパス式を用いたデータベース検索におけるアクセス権を判定する。

【選択図】 図2

認定・付加情報

| | |
|---------|----------------|
| 特許出願の番号 | 特願 2003-083243 |
| 受付番号 | 50300483383 |
| 書類名 | 特許願 |
| 担当官 | 小野寺 光子 1721 |
| 作成日 | 平成15年 5月 6日 |

<認定情報・付加情報>

【特許出願人】

| | |
|----------|---|
| 【識別番号】 | 390009531 |
| 【住所又は居所】 | アメリカ合衆国10504、ニューヨーク州 アーモンク ニュー オーチャード ロード |
| 【氏名又は名称】 | インターナショナル・ビジネス・マシーンズ・コーポレーション |

【代理人】

| | |
|----------|--|
| 【識別番号】 | 100086243 |
| 【住所又は居所】 | 神奈川県大和市下鶴間1623番地14 日本アイ・ビー・エム株式会社 大和事業所内 |
| 【氏名又は名称】 | 坂口 博 |

【代理人】

| | |
|----------|--|
| 【識別番号】 | 100091568 |
| 【住所又は居所】 | 神奈川県大和市下鶴間1623番地14 日本アイ・ビー・エム株式会社 大和事業所内 |
| 【氏名又は名称】 | 市位 嘉宏 |

【代理人】

| | |
|----------|--|
| 【識別番号】 | 100108501 |
| 【住所又は居所】 | 神奈川県大和市下鶴間1623番14 日本アイ・ビー・エム株式会社 知的所有権 |
| 【氏名又は名称】 | 上野 剛史 |

【復代理人】

| | |
|----------|--------------------------------------|
| 【識別番号】 | 100104880 |
| 【住所又は居所】 | 東京都港区赤坂5-4-11 山口建設第2ビル 6F セリオ国際特許事務所 |
| 【氏名又は名称】 | 古部 次郎 |

【選任した復代理人】

| | |
|--------|-----------|
| 【識別番号】 | 100118201 |
|--------|-----------|

次頁有

認定・付加情報（続き）

【住所又は居所】 東京都港区赤坂 5-4-11 山口建設第二ビル
6F セリオ国際特許事務所
【氏名又は名称】 千田 武

出 願 人 履 歴 情 報

識別番号 [390009531]

1. 変更年月日 2002年 6月 3日

[変更理由] 住所変更

住 所 アメリカ合衆国10504、ニューヨーク州 アーモンク ニ
ュー オーチャード ロード

氏 名 インターナショナル・ビジネス・マシーンズ・コーポレーショ
ン